Revisiting the Security of Approximate FHE with Noise-Flooding Countermeasures

Flavio Bergamaschi[1], Anamaria Costache[2], Dana Dachman-Soled[3], Hunter Kippen[3], Lucas LaBuff[3], and Rui Tang[3]

[1] Intel Labs
flavio@intel.com
[2] Norwegian University of Science and Technology
anamaria.costache@ntnu.no
[3] University of Maryland
{danadach@, hkippen@, llabuff@terpmail., ruitang@}umd.edu

Abstract. Approximate fully homomorphic encryption (FHE) schemes, such as the CKKS scheme (Cheon, Kim, Kim, Song, ASIACRYPT '17), are among the leading schemes in terms of efficiency and are particularly suitable for Machine Learning (ML) tasks. Although efficient, approximate FHE schemes have some inherent risks: Li and Micciancio (EUROCRYPT '21) demonstrated that while these schemes achieved the standard notion of CPA-security, they failed against a variant, $\mathsf{IND\text{-}CPA}^D$, in which the adversary is given limited access to the decryption oracle. Subsequently, Li, Micciancio, Schultz, and Sorrell (CRYPTO '22) proved that with noise-flooding countermeasures which add Gaussian noise of sufficiently high variance before outputting the decrypted value, the CKKS scheme is secure. However, the variance required for provable security is very high, inducing a large loss in message precision.
We consider a broad class of attacks on CKKS with noise-flooding countermeasures, which we call "semi-honest" attacks, in which an adversary obtains the view of an honest party who holds the public key and can make evaluation and decryption queries to an oracle. The ciphertexts submitted for decryption can be fresh ciphertexts, or ciphertexts resulting from the homomorphic evaluation of some circuit on fresh and independent ciphertexts. We analyze the concrete security of CKKS with various levels of noise-flooding in the face of such attacks. Our aim is to precisely quantify the various trade-offs between the number of allowed decryptions before key refreshing, noise-flooding levels, and the concrete security of the scheme after a number of decryptions have been observed. Due to the large dimension and modulus in typical FHE parameter sets, previous techniques even for *estimating* the concrete runtime of such attacks – such as those in (Dachman-Soled, Ducas, Gong, Rossi, CRYPTO '20) – become computationally infeasible, since they involve high dimensional and high precision matrix multiplication and inversion. We therefore develop new techniques that allow us to perform fast security estimation, even for FHE-size parameter sets.

Keywords: Approximate FHE · Concrete security · Average-case noise analysis.

1 Introduction

The notion of "approximate FHE" – fully homomorphic encryption schemes that guarantee only approximate correctness of decryption – was proposed by Cheon, Kim, Kim, and Song [17]. Their scheme, henceforth referred to as CKKS, is one of the leading schemes in terms of efficiency and Single Instruction/Multiple Data (SIMD) parallelization opportunities, and is particularly suitable for Machine Learning (ML) tasks. Although efficient, approximate FHE schemes have some inherent risks: Li and Micciancio [31] demonstrated what while these schemes achieved the standard notion of CPA-security, they failed against a variant, IND-CPA^D, in which the adversary is given limited access to the decryption oracle. In the same work [31], the authors showed that for exact schemes (such as BGV, BFV and TFHE), the notions of IND-CPA^D and IND-CPA are equivalent.[4]

Noise-flooding techniques have been suggested as a practical countermeasure against IND-CPA^D attacks [1]. These techniques add noise (from a Gaussian distribution) to the message obtained by decrypting a ciphertext, before it is returned to the adversary. Such countermeasures were formally analyzed in the work of Li, Micciancio, Schultz, and Sorrell [32], and it was shown that when the noise-flooding level is sufficiently high, they are indeed provably secure. Nevertheless, the amount of noise required for provable IND-CPA^D security remains quite high, and as a result, severely limits the message precision that CKKS can handle (8 or 16 bits of precision for parameter sets deemed "reasonable").

There are two main reasons for the large noise required for provable IND-CPA^D security. First, a *worst-case noise analysis* is needed to determine the amount of noise already present in a ciphertext prior to decryption and the noise flooding must then scale with this *worst-case* noise. The reason is that average-case noise analysis assumes that input ciphertexts to homomorphic circuits are independent and identically distributed. When a circuit computation is performed on correlated inputs (a simple example is adding a ciphertext to itself ℓ times instead of adding ℓ independent ciphertexts), the average-case analysis will fail to output the correct noise estimation. This particular correlated input attack has been exploited in [27]. A similar attack exploiting decryption failures in exact FHE schemes was presented in [15]. Second, the current techniques for proving IND-CPA^D show that the decryptions obtained in the two games of the indistinguishability definition are *statistically close*, whereas *computational indistinguishability* would be sufficient to achieve the security notion.

In this work, we introduce a formal security model that captures *semi-honest attackers with access to a decryption oracle*. As a result of enforcing semi-honest behavior, *average-case noise analysis* is sufficient to accurately estimate the noise present in a ciphertext submitted for decryption, so the noise-flooding level can

[4] In a recent work [15], this is called into question, as the authors point out that the proof of equivalence between IND-CPA^D and IND-CPA does not take into account the decryption failure probability of an exact scheme. The authors of [15] exploit the fact that this decryption failure probability is rather high in implementations of exact schemes to run an IND-CPA^D attack on the BFV scheme, and remark that their attack also applies to BGV and TFHE.

scale with the *average-case* noise, as opposed to the worst-case noise. We then investigate the *concrete computational security* achieved when decryption is augmented with different levels of noise flooding. In particular, we investigate the concrete runtime and success probability of the state-of-the-art key recovery attacks when incorporating the additional information obtained from decryption. Thus, our goal is to provide insights into the concrete security of CKKS with various noise-flooding levels in the semi-honest setting.

We consider internal threats, which are common in deployment scenarios, and argue that it is conceivable that adversaries access the internal randomness of the system, after honest parties have finished their computations. Our new model, which we call $\mathsf{IND\text{-}CPA}^{DSH}$, captures semi-honest attackers who do not hold the secret key, but may observe outputs of the decryption oracle and thus obtain noisy decryptions. In particular, our model enforces that public keys and ciphertexts are created honestly, and that evaluated ciphertexts correspond only to evaluations of *admissible* circuits[5], whose inputs correspond to fresh, independently generated ciphertexts, and where input ciphertexts are never re-used across evaluations. Our model captures an adversary who passively corrupts a party within the system and observes their entire state, and is incomparable to $\mathsf{IND\text{-}CPA}^{D}$. We note that in our model the adversary cannot choose the circuits to be evaluated adaptively, based on the internal randomness of the honest party. Instead, we assume the entire computation is performed honestly, and the adversary is only given the view of the honest party at the end of the experiment.

One may ask why we disallow the adversary from querying the decryption oracle with ciphertexts generated in ways other than the above. Indeed, when such queries are allowed, $\mathsf{IND\text{-}CPA}^{D}$ attacks against average-case noise-flooding techniques are known [27,15], as mentioned above. We note that in such attacks the attacker *actively* chooses the distribution over input ciphertexts or over the circuit to be evaluated such that the decrypted output deviates from the average-case distribution. On the other hand, we assume a "semi-honest" model, or alternatively, assume that external measures have been put in place in the particular deployment to ensure that an adversary cannot perform such attacks. For example, to ensure the integrity of the computation, as well as the well-formedness of the ciphertexts and relevant keys, Verifiable Computation (VC), and Zero-Knowledge (ZK) proofs for FHE schemes or even the use of enclaves should be considered [8,24,25]. The implementation of these measures are outside the scope of our work; we simply observe that if such measures are in place, then adversaries are restricted to be *semi-honest* as described above. See Section 4 for the formal security definition, as well as a comparison between our proposed model and the $\mathsf{IND\text{-}CPA}^{D}$ one.

Once we have established our security model, we investigate the best attacks in this model, for various levels of noise flooding. We consider the concrete security degradation of the CKKS scheme in the presence of t decryptions, with noise flooding of some variance ρ^2. Our starting point is noise-flooding equal to the

[5] In this work, admissible circuits will correspond to either identity, Class 1 or Class 2 circuits and will be defined subsequently.

noise level computed by the "average-case" noise analysis. Here, the decryption of a ciphertext is noise-flooded by the variance of the noise already present in the ciphertext[6]. This noise-flooding setting is optimal, in the sense that only 1 bit of message precision is lost. On the other side of the spectrum is setting ρ^2 as large as the variance needed for provable, statistical security. We investigate settings of ρ^2 that fall between these two extremes. Our aim is to present tradeoffs among (1) the number of allowed decryptions before the secret/public key must be refreshed, (2) the variance of the noise-flooding added to the decryption (which determines the loss of precision), and (3) the concrete security of the scheme after a number of decryptions have been observed by the adversary (e.g. a drop of 10 or 15 bits in security for a 256-bit parameter set may still be acceptable). We stress that the aim of our work is not to provide any definite conclusion on the concrete level of noise-flooding to apply when deploying CKKS. The conclusions of our experiments should rather be viewed as informing choices such as choosing parameter sizes and key refreshing policies.

Finally, our results are also applicable to threshold lattice-based encryption schemes [23,10,9,12]. In such schemes, partial decryptions of the form $\langle \mathtt{ct}, \mathtt{sk} \rangle = m + e$, for some noise e are released to all parties. These schemes are faced with the same issue as CKKS; the term e above contains information about the secret key. The typical approach is to noise-flood the partial decryption before broadcasting it – either with statistical noise flooding [10,12], or with some "lighter" noise flooding based on the Rényi divergence [14,18]. A recent work of Micciancio and Suhl [35] achieves security by noise flooding by very small amounts of noise (smaller than the noise already present in the ciphetext), but their techniques are not known to apply to the structured LWE setting.

In Section 2, we give a technical overview that includes a description of the classes of admissible circuits we consider, as well as our techniques for obtaining concrete hardness estimates. We emphasize that prior methods for concrete hardness estimation that are applicable in this setting, such as [21], require performing expensive matrix operations on the covariance matrix representing the conditional distribution of the LWE secret/error. For FHE-scale parameter sets, the covariance matrix can have dimension as high as $256K \times 256K$, so several hundred terabytes are required to naively store the values (assuming 64-bit precision, whereas in our experimental results in Section 9.2, we find that up to 2,000 bit precision is required for meaningful results). Therefore, one of our main technical contributions is developing new tools to provide fast and accurate estimates that do not require these high-dimensional matrix operations.

2 Technical Overview and Related Work

We consider three types of admissible circuits: *Identity*[7], *Class 1*, and *Class 2* circuits, which are defined below. For each type of circuit, we consider an

[6] As predicted by an average-case noise analysis [19].

[7] *Identity* circuits refer to fresh ciphertexts, i.e. ciphertexts on which *no homomorphic operation was performed*. This is the same terminology originally used by Li and Micciancio [31].

adversary who requests t evaluations of circuits of this type on fresh ciphertexts, and then obtains t noisy decryptions of these evaluations. Importantly, for each circuit type, the information obtained by the adversary from decryption will correspond to a noisy linear system of equations on the LWE secret and error underlying the CKKS public key. This means that the view of the adversary is equivalent to obtaining the public key $\mathtt{pk} = (-as + e, a) \pmod q$, for some ciphertext modulus q, along with a multivariate Gaussian distribution $\mathcal{N}(\mu', \mathbf{\Sigma}')$ representing the joint *conditional* distribution on the secret and error (s, e).

2.1 Admissible Circuits

Decryption Queries on Identity Circuits. We start by considering an attacker who submits a number t of fresh ciphertexts for decryption, or equivalently, requesting t decryptions of ciphertexts obtained from the evaluation of the identity circuit on a fresh encryption. The adversary receives the noise-flooded output of the decryption, where the noise is a centered Gaussian of some variance ρ^2. This is a natural circuit class to consider since in the original paper of Li and Micciancio [31], attacks using only identity circuits were shown to allow full key-recovery against CKKS *when there is no noise added during decryption.*

Decryption Queries on Class 1 and Class 2 Circuits. We then extend our analysis to broader classes of circuits, beyond identity circuits (see Section 8 for formal definitions of these classes). Briefly, Class 1 circuits are circuits that consist of ℓ independent subcircuits[8] $C_1, \ldots, C_\ell$. These circuits can be completely arbitrary as long as they all have the same multiplicative depth $d \geq 1$ and they each end in a *multiplication with rescale* operation. The final circuit consists of the addition of the outputs of these subcircuits. Intuitively, we require addition of ℓ ciphertexts so that the noise coefficients, which are individually uniformly random between $[-0.5, 0.5]$, can be well-approximated by independent Gaussian distributions. Class 2 circuits are circuits whose output corresponds to the *multiplication without rescale* of the outputs of two independent Class 1 circuits. Our motivation for considering Class 2 circuits is that in practice, a rescale is typically not performed in the final multiplication gate of the circuit, in order to reduce the size of the top-level modulus. We recall the noise terms for a multiplication with and without rescale in [11]. At a high level, the difference between the two is that the noise in the former is dominated by a rounding noise, whereas the latter contains more terms, including a quadratic equation in the secret decryption key.

For circuits in Class 1 and 2, we note that our attacker *does not need to know the internal randomness used by the encryption process*, and thus the attack is valid even in a weaker adversarial model. The analysis in this case is facilitated by the fact that it was shown in prior work [20,7] that after a **rescale** step, the *rounding* noise (which can be publicly computed) dominates the noise present in the ciphertext. We note that the same assumption was made in [32] and refer the

[8] By "independent" we mean that there are no wires crossing from one subcircuit to the other. We do not place any requirement on the distribution of the plaintext values corresponding to the inputs or outputs of the subcircuits.

reader to the paper for a further discussion. Upon decryption, the information obtained by the adversary corresponds to an approximate linear equation on the secret, which induces a conditional Gaussian distribution on the secret. Thus, the information obtained is in fact a special case of the information obtained by decryptions of the identity circuit, which corresponds to noisy linear equations on both the LWE secret and error.

2.2 Key Recovery Attacks

We consider three types of key recovery attacks for each of the three classes of admissible circuits described above. The attacks reduce the instance observed by the adversary to a unique-SVP (u-SVP) instance, which is the *same* approach used to determine the FHE parameter sets for varying levels of bit security in the first place! Our analysis differs in that we determine the effect of incorporating the knowledge that the LWE secret and error are jointly distributed as the multivariate Gaussian distribution $\mathcal{N}(\mu', \mathbf{\Sigma}')$ (capturing the conditional distribution on the LWE secret and error) on the concrete runtime of these attacks.

Lattice Reduction Attacks. Here we assume that the adversary embeds the original LWE instance and the distribution $\mathcal{N}(\mu', \mathbf{\Sigma}')$ into a *Distorted Bounded Distance Decoding* (DBDD) instance (introduced by [21]). Specifically, the resulting DBDD instance will consist of a tuple $(\Lambda, \mu', \mathbf{\Sigma}')$, where Λ is the lattice obtained by performing Kannan's embedding on the LWE instance $\mathtt{pk} = ([-as + e]_{q_L}, a)$ obtained from the CKKS public key (see Section 3.1 for more details). As shown by Dachman-Soled et al. [21], a DBDD instance can be reduced to a u-SVP instance, and solved using the state-of-the-art BKZ-algorithm. Using the terminology of [21], the information obtained by the adversary from decryption is denoted as "hints," and as discussed previously, these hints consist of noisy linear equations on the LWE secret/error, where the noise is sampled from a Gaussian distribution. Therefore, the conditional distribution on the LWE secret/error, given the hints, remains a Gaussian distribution and a closed-form formula for the new distribution can be obtained from known techniques. Thus, the steps to integrate the hints and transform the DBDD instance to a u-SVP instance follow those given in [21] for the case of *conditional, full-dimensional, approximate hints*. Upon obtaining the resulting u-SVP instance, the adversary then uses the BKZ algorithm to recover the shortest vector which corresponds to the LWE secret/error. As shown in [21], the time required by the BKZ algorithm in terms of bikz (i.e. BKZ-β) to solve the final u-SVP instance, can be accurately estimated given only the volume and dimension of the final u-SVP instance.

Importantly, although the attack template proceeds as the one outlined in [21], our *analysis* of the attack runtime differs. To obtain concrete security estimates for the runtime via full-dimensional approximate hints as in [21], one would need to compute the determinant of a $2n \times 2n$ dimensional matrix that depends on the t ciphertexts submitted for decryption and the outputs observed by the adversary. For $n = 256$ and $t = 16$, our experiments showed that this computation takes roughly a week on a supercomputer (See Section 9.2). In contrast, typical FHE parameters sets can have dimension up to $\log_2(n) = 17$. Thus,

to provide fast estimates, we analyze the *distribution* of the resulting $2n \times 2n$ dimensional matrix arising from the outlined attack. We provide a closed-form expression for the *expected* determinant of a matrix drawn from this distribution (See Section 5 and Lemma 5.1). We verify experimentally (See Section 9.2) that the predicted and actual expected determinant match closely. Further, to the best of our knowledge, ours is the first concrete analysis to crucially take into account the ring structure of the full-dimensional approximate "hints" obtained by the adversary. We believe this type of analysis is a crucial component for allowing concrete hardness estimates for FHE-size parameters.

Guessing Attacks. Here the attacker keeps track of the conditional multivariate Gaussian distribution on the LWE secret/error after integrating the t hints. When the variance of individual secret/error coordinates becomes small enough, the adversary rounds the coordinate of the mean of the multivariate Gaussian distribution to the nearest integer. At some point, the adversary can guess n out of $2n$ coordinates correctly with high probability, in which case it can solve the original LWE system to obtain the remaining n coordinates. Similarly to the lattice reduction case, actually keeping track of the covariance matrix of the multivariate Gaussian distribution requires a $2n \times 2n$ matrix inversion and is highly computationally intensive for FHE-scale parameters. Since we know the distribution of the matrix, we are able to derive bounds that hold with high probability on the trace and eigenvalues of the matrix, which in turn can be used to bound the success probability of the guessing attack, using the Gaussian correlation inequality [30] (See Section 6 and Lemma 6.1 for the case of identity circuits, and Section 8.3 for the case of Class 1 and Class 2 circuits).

Hybrid Attacks. Here the attacker guesses $g < n$ number of coordinates as above, but cannot guess n of them with sufficiently high probability. The attacker integrates these g guesses as "perfect hints" into the DBDD instance and finally obtains a new u-SVP instance, which it then solves using lattice reduction. After integrating the guesses, the information known to the adversary corresponds to a principal submatrix of the covariance matrix, whose determinant we need to compute in order to estimate hardness. As before, we do not compute the actual $2n \times 2n$ covariance matrix for the instance, which is highly computationally intensive, but rather use the fact that the distribution of the covariance matrix is known. We use the Eigenvalue Interlacing Theorem (see e.g. [28]) and bounds on the eigenvalues that hold w.h.p. in order to bound the determinant of the principal submatrix, given the determinant of the entire matrix (See Section 7 and Lemma 7.1 for the case of identity circuits, and Section 8.4 for the case of Class 1 and Class 2 circuits).

2.3 Summary of Experimental Results

We performed extensive experimentation for a wide range of parameter sets proposed by the `homomorphicencryption.org` standards [2], as well as a larger parameter set with a ring dimension of $\log_2 n = 17$ [33]. In Section 9, we provide experimental validation of Lemma 5.1, and in [11] we provide tables detailing the effectiveness of each of the three attack types on fresh ciphertexts (identity

circuits) at various noise-flooding levels: ρ^2_{circ}—the noise variance already present in a ciphertext—$100 \cdot \rho^2_{\mathsf{circ}}$, and $t \cdot \rho^2_{\mathsf{circ}}$, where t is the number of decryptions the attacker may observe[9]. We present the data for the analogous experiments on Class 1 and 2 circuits in the full version, see [11].

We note that our lemma statements involve complicated mathematical expressions for quantities such as the determinant or trace of the covariance matrix, and the implications for concrete security may not be immediately clear from these statements. The reason for this complexity is that in this work we strive for concrete (and in some cases achieve exact) values of the expected determinant or guessing probability, as opposed to asymptotic or approximate values. Further, our results are tailored to the ring-LWE setting which is crucially required by the CKKS scheme, and this setting introduces additional complexity as the entries of the matrices representing the noisy linear transformations of the secret and error are correlated instead of i.i.d. In order to obtain concrete estimates from the lemma and theorem statements, we ran scripts that used the expressions in the lemma and theorem statements, along with a BKZ-estimator, to compute the concrete hardness for various parameter sets and noise-flooding levels. As referenced above, we report our findings extensively in the full version [11].

In Section 10, we provide a graphical representation of our results and highlight our key findings. Most notably, we find that with noise-flooding levels of ρ^2_{circ} and $100 \cdot \rho^2_{\mathsf{circ}}$, full guessing attacks are feasible after observing a sufficient number of decryption queries (at most $\sim$ 100K needed), for all parameter sets and types of circuits considered. On the other hand, for noise level of $t \cdot \rho^2_{\mathsf{circ}}$, lattice reduction attacks are the only effective attacks.

Rephrasing the above, we investigate noise-flooding by $x \cdot \rho^2_{\mathsf{circ}}$, where x ranges from 1 to t, where t is the number of decryption queries. We recall that ρ^2_{circ} corresponds to the variance of the *average-case* noise that is already present in the ciphertext. It follows that noise-flooding by $x \cdot \rho^2_{\mathsf{circ}}$ incurs an *additional* loss of $\frac{1}{2}\log_2(x+1)$ bits in the message precision. This is in contrast to using the noise-flooding levels in [32], which incur a loss of an *additional* $\log_2(\sigma)+1$ bits of precision (beyond the *worst-case* noise already present in the ciphertext), where $\sigma = 8\sqrt{tn}2^{\kappa/2}$, κ is the security parameter, and n is the dimension (see Definition 18 and Theorem 3 of [32]).

2.4 Related Work

The inherent noise already present in a CKKS ciphertext was analyzed closely in [19]. We rely on their average-case analysis in our work in order to calibrate the noise-flooding noise and determine how much message precision is lost via the noise-flooding countermeasure.

The tools of incorporating side information on the LWE secret/error into a lattice reduction attack were developed in [21] via an introduction of an intermediate problem known as Distorted Bounded Distance Decoding (DBDD). Their framework allows the incorporation of "hints" into DBDD instances, which are finally converted to u-SVP instances via homogenization/isotropization, and

[9] Here circ denotes the circuit type that is being evaluated.

can be applied to analyze the concrete security of the CKKS scheme with noise-flooding countermeasures. However, in practice, keeping track of the intermediate DBDD instance is not feasible for FHE-scale parameters. The security estimation for the LWE problem was revisited in [22], but those techniques similarly do not scale to FHE-size parameter sets.

The work of Kim, Lee, Seo, and Song [29] considered the provable security of the Hint-LWE problem, and it can be observed that the information obtained from noisy decryptions of fresh ciphertexts can be viewed as an instance of Hint-LWE. Theorem 1 in [29] provides a security reduction from a spherical LWE instance to Hint-LWE. However, because the conditional Gaussian distribution arising from the Hint-LWE problem is ellipsoidal (not spherical), the reduction is not tight (additional noise is added to convert from the spherical to ellipsoidal distribution). This is in contrast to our approach, which provides an attack that first converts the Hint-LWE instance to a DBDD instance. Importantly, a DBDD instance with an ellipsoidal distribution is *equivalent* to another DBDD instance with a spherical distribution, and there is no loss in this reduction. Thus, our concrete security estimates are tighter, but only apply to certain classes of attack strategies. We also note that reduction in Theorem 1 of [29] is for decisional LWE, whereas our attacks are for the search LWE problem, making the two results somewhat incomparable.

Two recent works [27,15] present a key-recovery attack on the schemes CKKS and the exact FHE schemes, respectively. Both attacks rely on the following observation: an average-case noise analysis models all noise terms as independent Gaussians. When that assumption fails, the noise predicted by an average-case noise analysis will underestimate the actual noise observed. Indeed both works successfully run a key-recovery attack by using correlated inputs. We note that, while that research direction is interesting, this does not affect our setting. In particular, in all circuits we consider (the identity circuit, and the classes C1 and C2), the noise terms remain independent. We note that a recent work [5] argues that those attacks amount to incorrect estimation of the underlying ciphertext noise, as the heuristics specifically assume that inputs are independent, but [27,15] heavily rely on correlated inputs. The authors of [5] therefore define the notion of *application-aware* homomorphic encryption that can precisely counter these types of attacks. Our work therefore fits well within their model.

The work of Cheon, Hong and Kim [16] suggests noise-flooding countermeasures but does not delve deeply into the practical implications on the CKKS scheme's performance. Our work extends this by evaluating the practicality and efficiency of these countermeasures. Specifically, in our analysis, we examine the trade-offs between the number of allowed decryptions, noise-flooding levels, and concrete security. We show that while high levels of noise-flooding provide provable security, they significantly degrade message precision, making CKKS less practical for real-world applications (see [11]).

The work of Bootle, Delaplace, Espitau, Fouque and Tibouchi [13] discusses LWE problems under certain conditions, relevant to our discussion on lattice problems post-noise flooding. Specifically, their techniques are most relevant to

the analysis of our "guessing" attacks. They consider the case in which noisy linear equations (without reduction modulo q) are released on the LWE secret and error and provide bounds on the ratio of the noise versus coefficients of the linear equation needed to prevent guessing attacks. This corresponds to the setting of "approximate hints" that we use in this work to model the information learned by the adversary during decryption. Our results differ in that our analysis crucially takes into account the ring structure and distribution of the "hints" specific to the CKKS + noise-flooding setting. This is in contrast to [13] which assumes the hint vectors and the noise are independently drawn from distributions with known variances. Further, our goal is to provide a concrete, as opposed to asymptotic analysis. In particular, for a given CKKS + noise-flooding parameter set and a given target success probability, our analysis allows one to compute a concrete number of decryptions that are sufficient for the guessing attack to succeed with the target probability. Finally, our work derives the distribution of the "approximate hints" specific to the CKKS + noise-flooding setting and for the particular circuit classes we consider, whereas the prior work focused on the BLISS signature scheme.

The work of May and Nowakowski [34] shows a faster incorporation of hints into LWE problems, compared to that of Dachman-Soled et al. [21]. The reason we do not directly compare our efficiency to that of [34] is that their algorithms are only for *modular* and *perfect* hints (hints that correspond to noiseless linear equations modulo q or over the integers), whereas the hints required for the analysis in this work are *approximate hints*, which require computing the mean and covariance of a conditional Gaussian distribution.

Finally, Glaser, May, and Nowakowski [26] should be compared with our proposed technique, and we acknowledge that while it offers an efficient guessing method, our focus is on the practical complexity and concrete security estimates of such attacks in the context of CKKS. Our analysis includes the impact of noise-flooding on the effectiveness of guessing attacks and provides detailed estimates for the success probabilities of these attacks under various noise-flooding levels (see Section 6).

3 Preliminaries and Notation

Notation. We use bold lower case letters to denote vectors, and bold upper case letters to denote matrices. We use row notation for vectors, and denote by $\mathbf{I}_d$ the identity matrix of dimension d. We denote by $\{\mathbf{e}_i\}_{i\in[n]}$ the standard basis vectors in dimension n.

We use the notation R_q to denote the ring $\mathbb{Z}[x]/(\Phi_m(x), q)$, where $\Phi_m(x) = x^n + 1$, and $n = \phi(m)$ is a power of two. We denote ring elements by lowercase, non-bolded letters. When we employ a particular vector representation of a ring element in the coefficient or canonical embedding, we use vector notation. $[\cdot]_q$ denotes modular reduction $\pmod q$ (usually centered around 0).

We will make use of the canonical embedding and the subspace $H \subseteq \mathbb{C}^{\mathbb{Z}_m^*}$ defined as follows:

$$H = \{\mathbf{x} = (x_i)_{i\in\mathbb{Z}_m^*} \in \mathbb{C}^n : x_i = \overline{x_{-i}}, \forall i \in \mathbb{Z}_m^*\}.$$

H is isomorphic to $\mathbb{R}^n$ as an inner product space via the unitary transformation

$$B = \begin{pmatrix} \frac{1}{\sqrt{2}}\mathbf{I} & \frac{i}{\sqrt{2}}\mathbf{J} \\ \frac{1}{\sqrt{2}}\mathbf{J} & \frac{-i}{\sqrt{2}}\mathbf{I} \end{pmatrix}$$

where $\mathbf{I}$ is the identity matrix of size $n/2$ and $\mathbf{J}$ is its reversal matrix.

The canonical embedding of $a \in \mathbb{Q}[x]/\Phi_m(x)$ into $\mathbb{C}^n$ is the vector of evaluations of a at the roots of $\Phi_m(x)$. Specifically $\sigma(a) = [a(\zeta^j)_{j \in \mathbb{Z}_m^*}]$, where ζ is a primitive m-th root of unity. Due to the conjugate pairs, σ maps into the subspace H. When a is represented as a vector of coefficients $\mathbf{a}$, we can express the canonical embedding transformation as a linear transformation $\mathbf{aV}$.

We denote by $\mathcal{N}(\mu, \mathbf{\Sigma})$ the multivariate Gaussian with mean μ and covariance $\mathbf{\Sigma}$. We note that a multivariate Gaussian is fully determined by its mean and covariance. Thus, when the covariance of a dim dimensional multivariate Gaussian is a multiple of $\mathbf{I}_{\mathsf{dim}}$, the dim variables are all independent.

DBDD *and concrete hardness estimates.* A DBDD instance (defined in [21]) consists of a tuple $(\Lambda, \mu, \mathbf{\Sigma})$, where Λ is a lattice, and $(\mu, \mathbf{\Sigma})$ are viewed as the mean and covariance of a Gaussian distribution. Informally, the DBDD problem asks to find the unique vector in the lattice Λ that is contained in the ellipsoid defined by (μ, Σ) (for the formal definition see [21]). The prior work of [21] showed how to transform a DBDD instance into a u-SVP instance with lattice Λ' using the homogenization and isotropization steps, and further showed that the secret vector of this u-SVP instance has expected squared norm $||\mathsf{s}||^2 = \mathsf{dim}(\Lambda')$. Thus, standard techniques can be used to estimate the hardness of the resulting u-SVP instance, where hardness is measured in terms of the "bikz" or BKZ-β required to find the unique solution. In particular, following [3,6,21], β can be estimated as the minimum integer that satisfies

$$\sqrt{\beta} \leq \delta_\beta^{2\beta - \mathsf{dim}(\Lambda') - 1} \mathsf{Vol}(\Lambda')^{1/\mathsf{dim}(\Lambda')} \tag{1}$$

for a lattice Λ' where δ is the root-Hermite-Factor of BKZ-β.

The CKKS scheme. See [11] for a detailed description of the CKKS encryption scheme as well as a derivation of the error terms present in the message when decrypting a fresh ciphertext, and when decrypting after one or more multiplication steps (with or without a rescale operation). Following [19], we also present the noise variance in a fresh CKKS ciphertext, and in a ciphertext resulting from a multiplication and rescale operation (See [11]).

3.1 Modeling Noisy Decryptions as Hints

For the case of identity circuits (we will deal with attacks on Class 1 and Class 2 circuits in Section 8), we concretely consider an adversary who obtains a CKKS public key $\mathtt{pk} = ([-as + e]_q, a)$ and t independently sampled encryptions, and then asks for t decryptions of the constructed ciphertexts. Thus, for each $j \in [t]$, the adversary obtains the (noisy) polynomial $e_1^j \cdot s + v^j \cdot e \approx \gamma^j$, where multiplication is over the ring R_q. The adversary knows e_1^j and v^j whose coefficients

are modeled as independent Gaussians with 0 mean and variance $\sigma_{h_s}^2$ and $\sigma_{h_e}^2$, respectively. $(s||e)$ corresponds to the LWE secret/error used to construct the public key. Since we assume that all the polynomials involved have small magnitude, there is actually no wraparound modulo q. In this case, we can view the multiplication and addition as over the ring of integers $\mathbb{Z}[x]/\Phi_m(x)$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial of degree $n = \phi(m)$, and n is a power of two. Using the well-known representation of polynomial multiplication in $\mathbb{Z}[x]/\Phi_m(x)$ as matrix-vector multiplication with vectors in $\mathbb{Z}^n$, we note that decryptions correspond to "hints," or noisy linear systems of equations with respect to secret/error vectors $(\mathbf{s}||\mathbf{e})$. The matrices corresponding to these systems of linear equations can be combined into a single matrix denoted as $\mathbf{H}$ and referred to as the "hint matrix."

Since the original LWE secret/error distribution is (well-approximated) by a multivariate Gaussian (with mean $\mathbf{0}$ and covariance $\mathbf{\Sigma}$), and since the information obtained from decryption corresponds to noisy linear systems of equations on the LWE secret/error (where the noise is Gaussian), the information of the adversary after observing decryptions, is captured by a tuple $(\mathtt{pk}, \mu', \mathbf{\Sigma}')$, where $\mathtt{pk} = ([-as + e]_{q_L}, a)$ is an LWE instance, and $(\mu', \mathbf{\Sigma}')$ are the mean and covariance matrix of a multivariate Gaussian distribution corresponding to the *conditional distribution* of the LWE secret and error, given the information learned by the adversary during decryption (see, for example, Lemma 6 in [21]).

For purposes of our key recovery attacks, will further consider the related DBDD instance $(\mathbf{\Lambda}, \mu', \mathbf{\Sigma}')$, where $\mathbf{\Lambda}$ is the lattice obtained from the LWE instance via Kannan's embedding (see Section 3 and [21] for more details on the DBDD problem). Since the unique solution of this DBDD instance corresponds to the LWE secret and error, it is sufficient for our key recovery attacker to solve this DBDD instance. To estimate the concrete hardness of the DBDD instance $(\mathbf{\Lambda}, \mu', \mathbf{\Sigma}')$, it remains to compute $\mathsf{det}(\mathbf{\Sigma}')^{-1}$. In Section 5, we will not compute this quantity directly, but instead compute its expected value. While our result makes the simplifying assumption that the coordinates of $\mathbf{e}_1^j$ and $\mathbf{v}^j$ are Gaussian (as opposed to ternary distributions or discrete Gaussians), we crucially take into account the **ring-LWE** setting, which gives rise to the algebraic structure of the hint matrix $\mathbf{H}$. Indeed, the expected determinant would significantly differ in the standard LWE setting, where $\mathbf{H}$ would be modeled as a matrix whose entries are independent Gaussians. In our case, the entries of $\mathbf{H}$ are *correlated*, making the analysis more delicate.

4 Adversarial Model

For reasons of space, we present the definition of IND-CPA^D security in [11]. In this work, we introduce a modification of the IND-CPA^D that captures *semi-honest* attacks, in which the attacker passively corrupts a user in the system and obtains its view. We call our new notion IND-CPA^{DSH}, where SH stands for *Semi-Honest*. .

Definition 4.1 ($\mathsf{IND\text{-}CPA}^{DSH}$ Security with respect to admissible set $\mathcal{G}$)**.** *Let $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Encrypt}, \mathtt{Decrypt}, \mathtt{Eval})$ be a public-key homomorphic, approximate encryption scheme with plaintext space $\mathcal{M}$, ciphertext space $\mathcal{C}$, randomness space $\mathcal{R}$. Let the set $\mathcal{D}$ correspond to the image of $\mathtt{Decrypt}$. We define an experiment $\mathsf{Expr}_b^{\mathsf{indcpa}^{DSH}}[\mathcal{A}, \mathcal{G}]$, parametrized by a bit $b \in \{0,1\}$ and involving an efficient adversary $\mathcal{A}$, given access to the following oracles. The oracles share a common state $S \in (\mathcal{M} \times \mathcal{M} \times \mathcal{C} \times (\mathcal{R} \cup \{\perp\}) \times (\mathcal{D} \cup \{\perp\}), \{0,1\})^*$ consisting of a sequence of tuples. Each tuple consists of two messages, a ciphertext, the randomness used to generate the ciphertext or $\perp$, the decryption of the ciphertext or $\perp$, and a bit indicating whether the ciphertext is a fresh ciphertext that has not yet been included in an evaluation. The experiment is also parametrized by a set $\mathcal{G}$ that consists of admissible tuples (S, g, J), where S is a valid state, g is a function $g : \mathcal{M}^k \to \mathcal{M}$, and J is a sequence of indices $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$.*

- *An encryption oracle $\mathtt{Encrypt}(\mathtt{pk}, m_0, m_1)$ that, given a pair of plaintext messages m_0, m_1, computes $r \leftarrow \mathcal{R}$, $\mathtt{ct} = \mathtt{Encrypt}(\mathtt{pk}, m_b; r)$, and sets $d = \perp$, $u = 0$. If $m_0 \neq m_1$, it sets $\rho = \perp$ and if $m_0 = m_1$, it sets $\rho = r$. It extends the state*
$$S := [S; (m_0, m_1, \mathtt{ct}, \rho, d, u)].$$
- *An evaluation oracle $\mathtt{H}(\mathtt{evk}, g, J)$ that, given a function $g : \mathcal{M}^k \to \mathcal{M}$ and a sequence of indices $J = (j_1, \ldots, j_k) \in \{1, \ldots, |S|\}^k$, checks whether (S, g, J) is admissible by checking whether $(S, g, J) \in \mathcal{G}$. If so, the evaluation oracle computes the ciphertext $\mathtt{ct} \leftarrow \mathtt{Eval}(\mathtt{evk}, g, S[j_1].\mathtt{ct}, \ldots, S[j_k].\mathtt{ct})$, and extends the state*
$$S := [S; (g(S[j_1].m_0, \ldots, S[j_k].m_0), g(S[j_1].m_1, \ldots, S[j_k].m_1), \mathtt{ct}, \perp, \perp, 1)].$$
Additionally, for $\ell \in [k]$, it sets the j_ℓ-th tuple as follows
$$S[j_\ell] := (S[j_\ell].m_0, S[j_\ell].m_1, S[j_\ell].\mathtt{ct}, S[j_\ell].\rho, S[j_\ell].d, 1).$$
- *A decryption oracle $\mathtt{Decrypt}(\mathtt{sk}, j)$ that, given an index $j \leq |S|$, checks whether $S[j].m_0 = S[j].m_1$. If so, the decryption sets $d^* = \mathtt{Decrypt}(\mathtt{sk}, S[j].\mathtt{ct})$ and sets the j-th tuple of S as follows:*
$$S[j] := (S[j].m_0, S[j].m_1, S[j].\mathtt{ct}, S[j].\rho, d^*, 1).$$
- *At any point, the adversary can query its oracles with a special symbol $\bigstar$. When this occurs, the entire state S is returned to the adversary. After this point, no further queries can be made to any of the oracles.*

The experiment is defined as

$$\begin{aligned}\mathsf{Expr}_b^{\mathsf{indcpa}^{DSH}}[\mathcal{A}, \mathcal{G}](1^\kappa) :& (\mathtt{sk}, \mathtt{pk}, \mathtt{evk}) \leftarrow \mathtt{KeyGen}(1^\kappa)\\ & S := [\,]\\ & b' \leftarrow \mathcal{A}^{\mathtt{Encrypt}(\mathtt{pk},\cdot,\cdot), \mathtt{H}(\mathtt{evk},\cdot,\cdot), \mathtt{Decrypt}(\mathtt{sk},\cdot)}(1^\kappa, \mathtt{pk}, \mathtt{evk})\\ & \mathsf{return}(b')\end{aligned}$$

The advantage of adversary $\mathcal{A}$ with respect to admissible set $\mathcal{G}$ against the $\mathsf{IND\text{-}CPA}^{DSH}$ *security of the scheme is*

$$\mathsf{Adv}_{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}](\kappa) = \Big|\Pr[\mathsf{Expr}_0^{\mathsf{indcpa}^D}[\mathcal{A},\mathcal{G}](1^\kappa)=1] - \Pr[\mathsf{Expr}_1^{\mathsf{indcpa}^{DSH}}[\mathcal{A},\mathcal{G}](1^\kappa)=1]\Big|.$$

The scheme $\mathcal{E}$ is $\mathsf{IND\text{-}CPA}^{DSH}$*-secure with respect to admissible set $\mathcal{G}$ if for any efficient (probabilistic polynomial time) $\mathcal{A}$, the advantage* $\mathsf{Adv}_{\mathsf{indcpa}^D}[\mathcal{A},\mathcal{G}]$ *is negligible in κ.*

We mention some important points about the definition. First, note that the evaluation oracle checks whether the tuple $(S, g, J) \in \mathcal{G}$ and this check can take the stored u value into account – indicating whether a ciphertext is fresh and has not been previously inputted into an evaluation circuit – since it is part of the state S. This check is necessary for the validity of average-case noise analysis.

Second, our definition is non-adaptive and we note that this is also necessary for the validity of average-case analysis. For example, an adaptive adversary may query the encryption oracle to obtain a large number of fresh ciphertexts along with their randomness. Then it may choose a smaller subset of these ciphertexts with the highest noise and query the evaluation oracle with a circuit that performs a computation over these "biased" ciphertexts. In this case, average-case noise analysis will greatly underestimate the noise in the final decryption, since it inherently assumes the inputs are independent and distributed as fresh ciphertexts.

Importantly, we emphasize that the above attack also works in a model where the randomness of the ciphertexts *is not returned by the oracle* and evaluation of Class 1 or 2 circuits is allowed. This is because the dominating noise in Class 1 and 2 circuits comes from the "rounding error" during a homomorphic multiplication, and this error can be computed given knowledge of the input ciphertexts and evaluation key only. Thus, non-adaptivity is an inherent requirement for average-case noise analysis.

We provide a brief comparison with the $\mathsf{IND\text{-}CPA}^D$ model. Firstly, in the $\mathsf{IND\text{-}CPA}^D$ model, there is no (equivalent) value u checking for "freshness" of ciphertexts – and indeed, this was exploited in previous works to run an attack [15,27]. Secondly, our definition is non-adaptive, whereas in the $\mathsf{IND\text{-}CPA}^D$ model, the attacker has adaptive access to the evaluation and decryption oracles, enabling attacks such as those described above, where an attacker can actively bias the noise distribution of a ciphertext. In particular, we stress that simply adding the notion of admissible circuits to the $\mathsf{IND\text{-}CPA}^D$ notion does *not* enforce semi-honest behavior, whereas semi-honest behavior is enforced in the $\mathsf{IND\text{-}CPA}^{DSH}$ model. Finally, we release the encryption randomness to the adversary (when this does not lead to trivial distinguishing attacks), whereas $\mathsf{IND\text{-}CPA}^D$ does not.

We now explain how our attacks on identity circuits and Class 1/Class 2 circuits can be viewed as $\mathsf{IND\text{-}CPA}^{DSH}$ attacks.

The attack on identity circuits. Recall that our attacker simply asks for decryptions of fresh CKKS ciphertexts, and, given the internal randomness of the fresh ciphertexts and the *noisy* decryptions, runs a key recovery attack. We now formalize how this attack can be viewed as a $\mathsf{IND\text{-}CPA}^{DSH}$ attack. Our attacker will query the $\texttt{Encrypt}$ oracle t times with $m_0 = m_1 = 0$ and once with $m_0 \neq m_1$. It will then query the $\texttt{Eval}$ oracle t times with the identity function for each of the first t ciphertexts. Checking whether (S, g, J) is admissible corresponds to checking that g is the identity function, that J consists of a single index j and that $S[j].u = 0$. The adversary will make a decryption query for each of these t evaluated ciphertexts. The decryption oracle computes $\texttt{Decrypt}(\texttt{sk}, \texttt{ct}) + \mathcal{N}(0, \sigma_\epsilon^2)$, for some noise-flooding variance σ_ϵ^2. Our adversary will then query with $\bigstar$ to obtain the entire state S. and will use the information to run a full key recovery attack. Once it knows the key, it can trivially break indistinguishability by using the recovered key to decrypt the $(t+1)$-st ciphertext obtained from the encryption oracle and determine whether it encrypts m_0 or m_1.

The attack on Class 1 or Class 2 circuits. Recall that our attacker asks for decryptions of Class 1 or Class 2 circuits evaluated on fresh CKKS ciphertexts, and given the *noisy* decryptions, runs a key recovery attack. We now formalize how this attack can be viewed as a $\mathsf{IND\text{-}CPA}^{DSH}$ attack. The attacker first generates fresh ciphertexts corresponding to the inputs for t evaluations of the Class 1 or Class 2 circuit. It does this by querying the $\texttt{Encrypt}(\texttt{pk}, \cdot, \cdot)$ oracle sufficiently many times, where each call sets $m_0 = m_1$. It queries the $\texttt{Encrypt}(\texttt{pk}, \cdot, \cdot)$ a final time with $m_0 \neq m_1$. The $\texttt{H}_{\texttt{evk}}(\cdot, \cdot)$ oracle is then called with functions $g : \mathcal{M}^k \to \mathcal{M}$ in Class 1 or Class 2 and with input indices $J = (j_1, \ldots, j_k)$. Checking whether (S, g, J) is admissible corresponds to checking that g is in Class 1 or Class 2 and that for all $j \in J$, $S[j].u = 0$. Decryption queries are then made with ciphertexts $\texttt{ct}$ corresponding to the output of calls to $\texttt{H}(\texttt{evk}, \cdot, \cdot)$ as described above. The decryption oracle computes $\texttt{Decrypt}(\texttt{sk}, \texttt{ct}) + \mathcal{N}(0, \sigma_\epsilon^2)$, for some noise-flooding variance σ_ϵ^2. Our adversary will then query with $\bigstar$ to obtain the entire state S. and will use the information to run a full key recovery attack. Once it knows the key, it can trivially break indistinguishability by using the recovered key to decrypt the last ciphertext obtained from the encryption oracle and determine whether it encrypts m_0 or m_1.

5 Security Loss under a Lattice Reduction Attack

Recall that the matrix $\mathbf{\Sigma}$ corresponds to the original covariance matrix for the LWE secret and error. Formally, let $\mathbf{\Sigma}$ be an $2n \times 2n$ diagonal matrix with the first n diagonal entries set to σ_s^2, the second n diagonal entries set to σ_e^2. The matrix $\mathbf{\Sigma}_\varepsilon$ corresponds to the covariance of the noise in the set of linear equations obtained on the LWE secret $\mathbf{s}$ from decrypting a ciphertext. Formally, $\mathbf{\Sigma}_\varepsilon = \sigma_\varepsilon^2 \cdot \mathbf{I}_{tn}$. $\gamma = \gamma^1 || \cdots || \gamma^t$ corresponds to the obtained outputs.

First, note that for $j \in [t]$,

$$e_1^j \cdot s = \mathbf{sVBP} \left(\mathbf{M}(e_1^j) \right) \mathbf{P}^{-1} \mathbf{B}^{-1} \mathbf{V}^{-1},$$

where $\mathbf{V}$ is the canonical embedding transformation into $\mathbb{C}^n$, $\mathbf{B}$ is the matrix corresponding to the isomorphism between $H \subset \mathbb{C}^n$ and $\mathbb{R}^n$, $\mathbf{P}$ is a permutation matrix, and $\mathbf{A}_1^j := \mathbf{M}(e_1^j)$ is a block diagonal matrix with $n/2$ blocks, each of dimension 2×2, where the i-th block is

$$\mathbf{A}_{1,i}^j := \begin{bmatrix} 1/\sqrt{2}w_{i,h_s}^j & 1/\sqrt{2}w_{n-i,h_s}^j \\ -1/\sqrt{2}w_{n-i,h_s}^j & 1/\sqrt{2}w_{i,h_s}^j, \end{bmatrix}$$

and $\mathbf{w}_{h_s}^j = (w_{1,h_s}^j, \ldots, w_{n,h_s}^j)$ is equal to $\mathbf{w}_{h_s}^j = \mathbf{e}_1^j\mathbf{VB}$. Since $\mathbf{VB}$ is an isometry (an orthogonal matrix scaled by $\sqrt{n}$), we have that $\sigma_{h_s}^2(\mathbf{VB})(\mathbf{VB})^T = n\sigma_{h_s}^2 \cdot \mathbf{I}_n$. So the random variables $[w_{i,h_s}^j, w_{n-i,h_s}^j]_{j\in[t],i\in[n/2]}$ are distributed as independent Gaussians with variance $n\sigma_{h_s}^2$. Note that $\mathbf{R} = (\mathbf{VBP})$ is a real matrix, even though $\mathbf{V}$ and $\mathbf{B}$ themselves are complex.

Similarly, for $j \in [t]$,

$$v^j \cdot e = \mathbf{eVBP}\left(\mathbf{M}(\mathbf{v}^j)\right)\mathbf{P}^{-1}\mathbf{B}^{-1}\mathbf{V}^{-1}.$$

In this case, $\mathbf{A}_2^j := \mathbf{M}(v^j)$ is a block diagonal matrix with $n/2$ blocks, each of dimension 2×2, where the i-th block is

$$\mathbf{A}_{2,i}^j := \begin{bmatrix} 1/\sqrt{2}w_{i,h_e}^j & 1/\sqrt{2}w_{n-i,h_e}^j \\ -1/\sqrt{2}w_{n-i,h_e}^j & 1/\sqrt{2}w_{i,h_e}^j, \end{bmatrix}$$

and $\mathbf{w}_{h_e}^j = (w_{1,h_e}^j, \ldots, w_{n,h_e}^j)$ is equal to $\mathbf{w}_{h_e}^j = \mathbf{v}^j\mathbf{VB}$. Now for each $j \in [t]$, $i \in [n/2]$, w_{i,h_e}^j and w_{n-i,h_e}^j are random variables distributed as independent Gaussians with variance $n\sigma_{h_e}^2$.

Thus, if there are t decryption queries we can represent the hint matrix $\mathbf{H}$ as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1^1 & \mathbf{A}_1^2 & \ldots & \mathbf{A}_1^t \\ \mathbf{A}_2^1 & \mathbf{A}_2^2 & \ldots & \mathbf{A}_2^t \end{bmatrix} \begin{bmatrix} \mathbf{R}^{-1} & & \\ & \ddots & \\ & & \mathbf{R}^{-1} \end{bmatrix},$$

where $\mathbf{R}$ is an orthogonal matrix scaled by $\sqrt{n}$.

Applying the approximate hints of [21], the transformed covariance matrix $\mathbf{\Sigma}'$ and mean μ' are as follows (the dimension and lattice of the DBDD instance remain unchanged from the instance described in Section 3.1):

$$\mathbf{\Sigma}' = \mathbf{\Sigma} - \mathbf{\Sigma H}(\mathbf{H}^T\mathbf{\Sigma H} + \mathbf{\Sigma}_\varepsilon)^{-1}\mathbf{H}^T\mathbf{\Sigma} \tag{2}$$

$$\mu' = \gamma(\mathbf{H}^T\mathbf{\Sigma H} + \mathbf{\Sigma}_\varepsilon)^{-1}\mathbf{H}^T\mathbf{\Sigma}. \tag{3}$$

Our goal is to find $\det(\mathbf{\Sigma}')$. Given this, we can estimate the hardness of the new DBDD instance under a lattice reduction attack. However, instead of computing $\mathbf{\Sigma}'$ and then $\det(\mathbf{\Sigma}')$ exactly, which requires inversion of a $2n \times 2n$ matrix, we will instead compute the expected value of $\det(\mathbf{\Sigma}')$, where the expectation is taken over the choice of the hint matrix $\mathbf{H}$.

Using a generalization of the Matrix Determinant Lemma, we obtain:

$$\mathbb{E}[\det((\mathbf{\Sigma}')^{-1})] = \mathbb{E}\left[\frac{\det(\mathbf{H}^T\mathbf{\Sigma}\mathbf{H}+\mathbf{\Sigma}_\varepsilon)}{\det(\mathbf{\Sigma}_\varepsilon)\det(\mathbf{\Sigma})}\right]. \tag{4}$$

Since $\mathbf{\Sigma}_\varepsilon$ and $\mathbf{\Sigma}$ are diagonal matrices whose entries depend on the parameters of the FHE cryptosystem, their determinants are constants and are easy to compute. Thus, it remains to compute $\mathbb{E}[\det(\mathbf{H}^T\mathbf{\Sigma}\mathbf{H}+\mathbf{\Sigma}_\varepsilon)]$, which can then be plugged into (4).

Lemma 5.1. *Let* $\mathbf{H}, \mathbf{R}, [\mathbf{A}_1^j = \mathbf{M}(e_1^j), \mathbf{A}_2^j = \mathbf{M}(v^j)]_{j\in[t]}$ *be as described above. Then*

$$\begin{aligned}\mathbb{E}[\det(\mathbf{H}^T\mathbf{\Sigma}\mathbf{H}+\mathbf{\Sigma}_\epsilon)] = \\ \Big(\sigma_s^4\sigma_e^4\sigma_\epsilon^{4t-8}\Big(\frac{7}{4}t(t-1)n^4\sigma_{h_s}^4\sigma_{h_e}^4 + tn^2\sigma_\epsilon^4(\frac{\sigma_{h_s}^4}{\sigma_e^4}+\frac{\sigma_{h_e}^4}{\sigma_s^4}) \\ + \Big(t(t-1)n^2\sigma_{h_s}^2\sigma_{h_e}^2 + tn\sigma_\epsilon^2(\frac{\sigma_{h_s}^2}{\sigma_e^2}+\frac{\sigma_{h_e}^2}{\sigma_s^2}) + \frac{\sigma_\epsilon^4}{\sigma_s^2\sigma_e^2}\Big)^2\Big)\Big)^{\frac{n}{2}},\end{aligned}$$

where the expectation is taken over choice of $\mathbf{e}_1^j \sim \mathcal{N}(0,\sigma_{h_s}^2)^n$ *and* $\mathbf{v}^j \sim \mathcal{N}(0,\sigma_{h_e}^2)^n$ *for all* $j \in [t]$.

The proof can be found in [11].

Obtaining the final hardness estimates. One can perform homogenization/isotropization of the DBDD instance (as in [21]) to obtain a u-SVP instance and then estimate the BKZ-β for that instance. However, as described in [21], one can obtain the BKZ-β estimates using only the dimension and volume of the lattice after homogenization/isotropization, and the lattice basis itself is not required. The lattice in our DBDD instance is a q_L-ary lattice and thus has log volume $n\cdot\ln(q_L)$. After homogenization/isotropization, the log volume of the lattice increases to $n\ln(q_L)+\ln(\det((\mathbf{\Sigma}')^{-1}))/2$. Using (4) and Lemma 5.1, we use the expectation of $\det((\mathbf{\Sigma}')^{-1})$ in the above formula. The dimension remains unchanged after integrating hints. Thus, this information is sufficient for obtaining BKZ-β estimates for the final u-SVP instance.

6 Key Recovery via Guessing

When $\mathbf{\Sigma}'$ in (2) has sufficiently small variance, then instead of running a lattice reduction attack, another strategy is to simply guess coordinates of the LWE secret/error by rounding the mean μ' in (3) to the nearest integer. If n coordinates of these coordinates are guessed and all guesses are correct, then the entire LWE secret/error can be recovered by solving a linear system modulo q. To analyze the success of the above attack we begin with the following lemma:

Lemma 6.1. *Let* $\mathbf{\Sigma}'$ *be defined as in (2). Then* $\mathbf{Tr}(\mathbf{\Sigma}') \le \mathsf{T} = n\cdot\frac{\left(\frac{\sigma_s^2\cdot\sigma_e^2\cdot 2t\cdot n(\sigma_{h_s}^2+\sigma_{h_e}^2)}{2\cdot\sigma_\epsilon^2}+\sigma_s^2+\sigma_e^2\right)}{B} + \frac{3\sqrt{2n\cdot V}}{B}$ *with probability at least* $0.99 - 3n\cdot e^{-12.25}$

over choice of hint vectors, where

$$B = \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t - 7\sqrt{2t})^2 \cdot n^2 \sigma_{h_s}^2 \cdot \sigma_{h_e}^2}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot (2t - 7\sqrt{2t}) \cdot n\sigma_{h_s}^2}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_e^2 \cdot (2t - 7\sqrt{2t})(n\sigma_{h_e}^2)}{2 \cdot \sigma_\epsilon^2} + 1 - \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot n^2(\sigma_{h_s}^2 + \sigma_{h_e}^2)^2(3.5\sqrt{2t} + 12.25)^2}{2 \cdot \sigma_\epsilon^4}$$

$$V = \frac{\sigma_s^4 \cdot \sigma_e^4 \cdot (\mathbb{E}[R_1^2] + \mathbb{E}[R_2^2])}{4 \cdot \sigma_\epsilon^4} + 2\frac{\sigma_s^4 \cdot \sigma_e^4 \cdot \mathbb{E}[R_1] \cdot \mathbb{E}[R_2]}{4 \cdot \sigma_\epsilon^4} + 2\frac{(\sigma_s^4 \cdot \sigma_e^2 + \sigma_s^2 \cdot \sigma_e^4) \cdot (\mathbb{E}[R_1] + \mathbb{E}[R_2])}{2 \cdot \sigma_\epsilon^2} + (\sigma_s^2 + \sigma_e^2)^2 - \left(\frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_1]}{2 \cdot \sigma_\epsilon^2} + \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot \mathbb{E}[R_2]}{2 \cdot \sigma_\epsilon^2} + \sigma_s^2 + \sigma_e^2\right)^2$$

$$\mathbb{E}[R_1] = 2t \cdot n\sigma_{h_s}^2$$
$$\mathbb{E}[R_2] = 2t \cdot n\sigma_{h_e}^2$$
$$\mathbb{E}[R_1^2] = 4tn^2\sigma_{h_s}^4 + 4t^2n^2\sigma_{h_s}^4$$
$$\mathbb{E}[R_2^2] = 4tn^2\sigma_{h_e}^4 + 4t^2n^2\sigma_{h_e}^4$$

We note that up to parameter setting of $n = 32768$, the success probability in the above claim is at least 0.52.[10] The proof of the lemma can be found in [11].

Given the above, we consider the distribution of $\mathbf{e}||\mathbf{s}-\mu'$, where μ' is the mean from equation (3). The random variable $\mathbf{e}||\mathbf{s}-\mu'$ is distributed as the multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}')$. μ' is the correct guess for $\mathbf{e}||\mathbf{s}$ as long as for all $i \in [n]$ $|e_i - \mu'_i| \leq 0.5$ and for all $i \in [n]$ $|s_i - \mu'_{i+n}| \leq 0.5$. The probability that the above occurs for each coordinate is the same as the probability weight of the hypercube corresponding to $-0.5 \leq x_i \leq 0.5, i \in [n]$ under the multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma}')$. We use the following theorem to lower bound this probability weight:

Theorem 6.2 (Special case of the Gaussian Correlation Inequality [30]). *Let* $\mathbf{X}$ *be an* n*-dimensional Gaussian random variable. Then for any* $t_1, \ldots, t_n > 0$,

$$\mathbb{P}(|X_1| \leq t_1, \ldots, X_n \leq t_n] \geq \mathbb{P}(|X_1| \leq t_1) \cdots \mathbb{P}(|X_n| \leq t_n].$$

We instantiate the above theorem with $\mathbf{X}$ consisting of a subset S of size n of the coordinates of the conditional Gaussian distribution $((\mathbf{s}||\mathbf{e})-\mu') \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}')$, with $t_j = 0.5, j \in S$ We thus have that

$$\mathbb{P}(|X_j| \leq t_j, i \in S) \geq \Pi_{j \in S} \mathbb{P}_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \mathbf{\Sigma}' \mathbf{e}_j^T)}(|X_j| \leq t_j], \tag{5}$$

where the $\mathbf{e}_j$ are the standard basis vectors.

[10] For the parameter sets with $n = 131072$, we increase 7 to 7.5, 3.5 to 3.75, 12.25 to 14.0625, and increase the probability to $0.99 - 3n \cdot e^{-14} > 0.66$.

To analyze $\Pr_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T)}[X_j \leq 0.5]$, we note that $\sum_{i \in [2n]} \mathbf{e}_i \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_i^T = \mathsf{Tr}(\mathbf{\Sigma}')$. By Lemma 6.1, we have that $\mathsf{Tr}(\mathbf{\Sigma}') \leq \mathsf{T}$ with 53% probability. Let $S \subseteq [2n]$ of size n be the set of indices j corresponding to the n smallest values among $\{\mathbf{e}_i \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_i^T : i \in [2n]\}$ this set of minimum values. Using the analysis in [11], we have that for each $j \in S$, $\mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T \leq \frac{\mathsf{T}}{2n}$. Therefore,

$$\Pr_{X_j \sim \mathcal{N}(0, \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T)}[|X_j| \leq 0.5] \geq -\mathsf{erf}\left(\frac{-0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right) = \mathsf{erf}\left(\frac{0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right). \tag{6}$$

Finally, the attack is as follows: The adversary chooses to guess the values of $\mathbf{e}_j$ or $\mathbf{s}_j$ for these n smallest values (corresponding to the set S), and then use the LWE instance to solve for the remaining n variables. The probability that all of the adversary's guesses are correct is lower bounded by the probability weight on the hypercube corresponding to $|X_j| \leq 0.5, j \in I$ when X is drawn from the multivariate Gaussian distribution $X \sim \mathcal{N}(0, \mathbf{\Sigma}')$. Using (5) and (6), this is at most

$$\prod_{j \in S} -\mathsf{erf}\left(\frac{-0.5}{\sqrt{2 \cdot \mathbf{e}_j \cdot \mathbf{\Sigma}' \cdot \mathbf{e}_j^T}}\right) \geq \left(-\mathsf{erf}\frac{-0.5}{\sqrt{2 \cdot \frac{\mathsf{T}}{2n}}}\right)^n = \mathsf{erf}\left(\frac{0.5}{\sqrt{\frac{\mathsf{T}}{n}}}\right)^n.$$

The final success probability of the attack is:[11]

$$\mathsf{erf}\left(\frac{0.5}{\sqrt{\frac{\mathsf{T}}{n}}}\right)^n - 3n \cdot e^{-12.25} - 0.01. \tag{7}$$

7 Hybrid Guessing/Lattice-Reduction Attacks

Recall the structure of the eigenvalues of $\mathbf{\Sigma}'$: There are $[n/2]$ blocks and for each $i \in [n/2]$, the eigenvalues $(\alpha_{4i+1}, \alpha_{4i+2}, \alpha_{4i+3}, \alpha_{4i+4})$, where $\alpha_{4i+1} = \alpha_{4i+3}$, $\alpha_{4i+2} = \alpha_{4i+4}$. For each $i \in [n/2]$, we say that $\{\alpha_{4i+1}, \alpha_{4i+2}\}$ and $\{\alpha_{4i+3}, \alpha_{4i+4}\}$ are pairs. For each i, the adversary computes $\mathbf{e}_i \mathbf{\Sigma}' \mathbf{e}_i^T$ and guesses μ_i for the g minimum values where g is the maximum value such that

$$\mathsf{erf}\left(\frac{0.5}{\sqrt{\frac{\mathsf{T}}{n}}}\right)^g \geq p, \tag{8}$$

for some threshold p. These guesses are made and incorporated as perfect hints. After this process, the covariance matrix is a principal submatrix of $\mathbf{\Sigma}'$ of dimension $(2n-g) \times (2n-g)$, which we denote by $\mathbf{\Sigma}''$. We denote by $\mathsf{PSub}_{2n-g}(\mathbf{\Sigma}')$ the set of all principal submatrices of $\mathbf{\Sigma}'$ of dimension $2n-g$. Similarly, the lattice reduces dimension by g and its volume remains the same. The following lemma gives a bound on the determinant of $\mathbf{\Sigma}''$.

[11] And for $n = 131072$, we replace $e^{-12.25}$ with e^{-14}.

Lemma 7.1. *Let $g \in \{0, 1, \ldots, n\}$. Let $\boldsymbol{\Sigma}'$ be defined as in (2). Let $\boldsymbol{\Sigma}'' = \mathsf{argmax}_{\widetilde{\boldsymbol{\Sigma}} \in \mathsf{PSub}_{2n-g}(\boldsymbol{\Sigma}')} \mathsf{Tr}(\widetilde{\boldsymbol{\Sigma}})$. With probability $0.99 - 4n \cdot e^{-12.25}$ over choice of hint vectors,*[12]

$$\mathbf{Tr}(\boldsymbol{\Sigma}') \leq \mathsf{T} \quad \textit{and} \quad \mathsf{det}(\boldsymbol{\Sigma}'') \leq \frac{\mathsf{det}(\boldsymbol{\Sigma}')}{\left(\frac{L}{U}\right)^g},$$

where T and B are defined as in Lemma 6.1, and

$$\begin{aligned}
L &= \frac{G + \sqrt{G^2 - 4 \cdot B \cdot \sigma_s^2 \cdot \sigma_e^2}}{2 \cdot B} \\
U &= \frac{\sigma_s^2 \cdot \sigma_e^2}{B_{max}} \\
G &= \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_e}^2) 2 \cdot \sigma_\epsilon^2 \\
&\quad + \sigma_s^2 \cdot \sigma_e^2 (2t + 7\sqrt{2t} + 24.5) \cdot (n\sigma_{h_s}^2) 2 \cdot \sigma_\epsilon^2 + \sigma_s^2 + \sigma_e^2 \\
B_{max} &= \frac{\sigma_s^2 \cdot \sigma_e^2 \cdot (2t + 7\sqrt{2t} + 24.5)^2 \cdot n^2 \sigma_{h_s}^2 \cdot \sigma_{h_e}^2}{4 \cdot \sigma_\epsilon^4} + \frac{\sigma_s^2 \cdot (2t + 7\sqrt{2t} + 24.5) \cdot n\sigma_{h_s}^2}{2 \cdot \sigma_\epsilon^2} \\
&\quad + \frac{\sigma_e^2 \cdot (2t + 7\sqrt{2t} + 24.5)(n\sigma_{h_e}^2)}{2 \cdot \sigma_\epsilon^2} + 1.
\end{aligned}$$

The proof of the lemma can be found in [11].
Combining Lemma 6.1 with Theorem 6.2 as before, we estimate that with at least $p - 4n \cdot e^{-12.25} - 0.01$ probability, all g number of guesses are correct, and

$$\mathsf{det}(\boldsymbol{\Sigma}'') \leq \frac{\mathsf{det}(\boldsymbol{\Sigma}')}{\left(\frac{L}{U}\right)^g}. \tag{9}$$

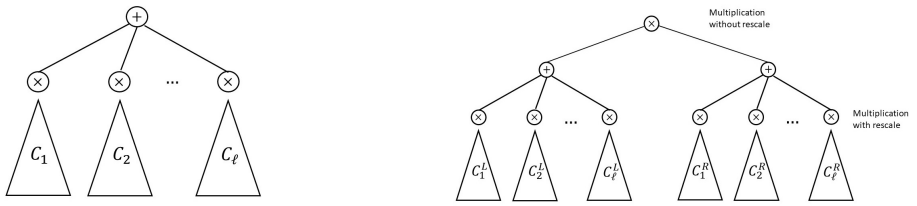
We note that for up to $n = 32768$, $4n \cdot e^{-12.5} \leq 0.63$. [13] As before, $\mathbb{E}[\mathsf{det}(\boldsymbol{\Sigma}')]$ can be computed via Lemma 5.1. Thus, we can use (9) to obtain a bound on the expected value of $\mathsf{det}(\boldsymbol{\Sigma}'')$ (conditioned on events with probability at least $0.99 - 4n \cdot e^{-12.25}$ occurring), compute the log-volume of the lattice after homogenization/isotropization as described in Section 3.1, and use the log-volume and dimension to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.

8 Extending to Larger Classes of Circuits

8.1 The First Class of Circuits and Lattice Reduction Attacks

In Figure 1a we present the first class of circuits we consider. The circuits $C_1, \ldots, C_\ell$ that are depicted each consist of $\log(r)$ levels of multiplications as well as any number of additions. The final gate in each of the circuits $C_1, \ldots, C_\ell$ is a multiplication with rescale. Note that the noise after multiplication with

[12] For the parameter sets with $n = 131072$, we increase 7 to 7.5, 24.5 to 28.125 and increase the probability to $0.99 - 4n \cdot e^{-14}$.
[13] And for $n = 131072$, $4n \cdot e^{-14} \leq 0.44$.



(a) First Class of Circuits

(b) Second Class of Circuits

Fig. 1: A pictorial representation of the two classes of circuits we consider.

rescale in circuit C_i is dominated by $\delta_1^i \cdot s + \delta_0^i$ (see [11]), where δ_0^i, δ_1^i are distributed as uniform random variables in the range $[-0.5, 0.5]$.

The final gate of the entire circuit is an addition gate that adds the outputs of each of the C_i circuits. We require ℓ subcircuits and a final addition gate in order to ensure that the linear coefficients of the noise polynomial (which are independent and uniform random in the range $[-0.5, 0.5]$ for each of the ℓ circuits) can be approximated by Gaussian random variables with mean 0 and variance $\frac{\ell}{12}$, the setting for which our Lemma 5.1 applies.

Specifically, the lattice reduction attack for circuits of this class can be analyzed by instantiating Lemma 5.1 with the following parameter settings.

- $\sigma^2_{h_s} = \frac{\ell}{12}$
- $\sigma^2_{h_e} = 0$
- σ^2_ϵ is set to the noise-flooding noise. The variance of the noise already present in the ciphertext can be computed by taking the noise in each ciphertext before addition (which [11] provides) and multiplying by ℓ.

8.2 The Second Class of Circuits and Lattice Reduction Attacks

In Figure 1b we present the second class of circuits we consider. The circuits $C_1^L, \ldots, C_\ell^L, C_1^R, \ldots, C_\ell^R$ that are depicted each consist of $\log(r)$ levels of multiplications as well as any number of additions. The final gate in each of the circuits $C_1^L, \ldots, C_\ell^L, C_1^R, \ldots, C_\ell^R$ is a multiplication with rescale. Note that the noise after multiplication with rescale in circuit C_i^L (resp. C_i^R) is dominated by $\delta_1^{L,i} \cdot s + \delta_0^{L,i}$ (resp. $\delta_1^{R,i} \cdot s + \delta_0^{R,i}$) (see [11]), where $\delta_0^{L,i}, \delta_1^{L,i}$ (resp. $\delta_0^{R,i}, \delta_1^{R,i}$) are distributed as uniform random variables in the range $[-0.5, 0.5]$. Thus, after the summation gates on the second level from the top, the linear and constant coefficients of the noise corresponding to the left and right summations can be approximated by Gaussian random variables $G_{L,1}, G_{L,0}, G_{R,1}, G_{R,0}$ with mean 0 and variance $\frac{\ell}{12}$.

These outputs are then multiplied via a multiplication *without rescale* gate. For most parameter settings, the dominating terms of the error after the final multiplication without rescale will correspond to $\frac{m^r}{\Delta^{r-1}} \cdot (G_{L,1} + G_{R,1}) \cdot s$. Further, the dominating linear coefficients of s are again (well approximated by) a

Gaussian of variance $\sigma_{h_s}^2 = \frac{\ell}{6} \cdot (\frac{m^r}{\Delta^{r-1}})^2$. Since the error term does not include information about e, we can set $\sigma_{h_e}^2 = 0$.

We compute the noise variance that is already present in the ciphertext, as a contribution of the following terms $\frac{m^r}{\Delta^{r-1}} \cdot (G_{L,0} + G_{R,0})$, $\frac{m^r}{\Delta^{r-1}} \cdot (G_{L,1} + G_{R,1}) \cdot s$, $(G_{L,1} \cdot G_{R,1}) \cdot s^2$, $(G_{L,0} \cdot G_{R,1}) \cdot s$, $(G_{L,1} \cdot G_{R,0}) \cdot s$, $G_{L,0} \cdot G_{R,0}$. Since the covariance of the above terms is 0, the total variance is the sum of the variances each term above. For the derivation of the variance of each of the above terms, see [11]. The total noise in the ciphertext has variance:

$$2\left(\frac{m^r}{\Delta^{r-1}}\right)^2 \cdot \frac{\ell}{12} \cdot \sigma_s^2 + \left(\frac{m^r}{\Delta^{r-1}}\right)^2 \cdot \frac{n \cdot \ell}{6} \cdot \sigma_s^2 + \frac{5}{2} n^3 \left(\frac{\ell}{12}\right)^2 \sigma_s^4 + 2n^2 \left(\frac{\ell}{12}\right)^2 \sigma_s^2 + n \cdot \left(\frac{\ell}{12}\right)^2$$

Obtaining the hardness estimates. We can now apply Lemma 5.1 with the following parameter settings:

- $\sigma_{h_s}^2 = \frac{\ell}{6} \cdot (\frac{m^r}{\Delta^{r-1}})^2$
- $\sigma_{h_e}^2 = 0$
- σ_ϵ^2 is set to the noise-flooding noise plus an additional $\frac{5}{2} n^3 \left(\frac{\ell}{12}\right)^2 \sigma_s^4 + 2n^2 \left(\frac{\ell}{12}\right)^2 \sigma_s^2$, the noise from the quadratic terms and the linear but non-Gaussian terms (which comes from the terms of the form $(G_{L,0} \cdot G_{R,1}) \cdot s$).

Note that the noise-flooding noise has variance at least $(\frac{m^r}{\Delta^{r-1}})^2 \cdot \frac{n \cdot \ell}{6} \cdot \sigma_s^2$, since the noise already in the ciphertext is larger than this quantity. Thus, for $n \in \mathbb{N}$, when

$$(\frac{m^r}{\Delta^{r-1}})^2 \gg \frac{5}{2} n^2 \cdot \frac{\ell}{24} + 2 \cdot n \cdot \frac{\ell}{24} > \frac{9}{2} n^2 \cdot \frac{\ell}{24}, \tag{10}$$

and m achieves the maximum allowed magnitude B_{msg} of each coordinate in the *encoded* plaintext (in which the message is viewed as a vector in the canonical embedding and is scaled up by Δ), we have that the noise-flooding noise dominates the additional $\frac{5}{2} n^3 \left(\frac{\ell}{12}\right)^2 \sigma_s^4 + 2n^2 \left(\frac{\ell}{12}\right)^2 \sigma_s^2$. Typically, after encoding, the maximum allowed magnitude of m in the canonical embedding is $\approx \Delta$. Thus, (10) is satisfied when $\Delta \geq \frac{3n}{4} \cdot \sqrt{\frac{\ell}{3}}$, which is typically satisfied for most parameter settings (in fact, Δ is typically far larger).

Thus, we can plug the above parameter settings into Lemma 5.1 to obtain the hardness estimates for these circuits under a lattice reduction attack.

8.3 Guessing Attack for Class 1 and 2 Circuits

Now that we have determined $\sigma_{h_s}^2$, $\sigma_{h_e}^2$, and σ_ϵ^2 for Class 1 and Class 2 circuits, we can use those values to derive formulas for the concrete security for guessing and hybrid attacks as well.

Recall that for Class 1 and Class 2 circuits, the hints are only on the $\mathbf{s}$ coordinates. So $\mathbf{\Sigma}'$ is a block matrix where the lower right hand $n \times n$ submatrix is a diagonal matrix with diagonal $(\sigma_e^2, \ldots, \sigma_e^2)$ and the upper left hand $n \times n$ submatrix has n eigenvalues of the form $[(\alpha_{2i+1}, \alpha_{2i+2})]_{i \in [n/2]}$ and for all $i \in$

$[n/2]$, $\alpha_{2i+1} = \alpha_{2i+2}$. Further, for each $i \in [n/2]$,

$$\alpha_{2i+1} = \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 \cdot R_{1,i}}{2\sigma_\epsilon^2}}.$$

Since with all but e^{-11} probability[14], $R_{1,i} \geq (2t - 6.63\sqrt{2t}) \cdot n\sigma_{h_s}^2$, we have that with probability $1 - n/2 \cdot e^{-11}$ all eigenvalues are less than

$$\sigma_{max}^2 \leq \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 \cdot (2t - 6.63\sqrt{2t}) \cdot n\sigma_{h_s}^2}{2\sigma_\epsilon^2}}, \tag{11}$$

and so for every standard basis vector $\mathbf{e}_i$, $\mathbf{e}_i \boldsymbol{\Sigma}'_S \mathbf{e}_i^T \leq \sigma_{max}^2$.

Finally, using the same techniques as above, this means that the guessing probability is at least

$$\mathsf{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^n. \tag{12}$$

Thus the total success probability of the attack is $\mathsf{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^n - n/2 \cdot e^{-11}$. We note that for up to parameter $n = 32768$, $n/2 \cdot e^{-11} \leq 0.28$.[15]

8.4 Hybrid Attack for Class 1 and 2 Circuits

Again, the attack for both Class 1 and Class 2 circuits is the same, with the only difference being the settings of $\sigma_{h_s}^2$, $\sigma_{h_e}^2$, and σ_ϵ^2 in the two cases.

The guessing strategy for the hybrid attack is as follows: For each i, the adversary computes $\mathbf{e}_i \boldsymbol{\Sigma}' \mathbf{e}_i^T$ and guesses μ_i for the g number of indices i with the minimum values of $\mathbf{e}_i \boldsymbol{\Sigma}' \mathbf{e}_i^T$, where g is the maximum value such that

$$\mathsf{erf}\left(\frac{0.5}{\sqrt{2\sigma_{max}^2}}\right)^g \geq p, \tag{13}$$

for some probability threshold p. These guesses are made and incorporated as perfect hints. After this process, the covariance matrix is a principal submatrix of $\boldsymbol{\Sigma}'_S$ of dimension $(n-g) \times (n-g)$, which we denote by $\boldsymbol{\Sigma}''_S$. Similarly, the lattice reduces dimension by g and its volume remains the same.

Let $\alpha_1, \ldots, \alpha_g$ be the g minimum eigenvalues of $\boldsymbol{\Sigma}'_S$. Using the Eigenvalue Interlacing Theorem [28], we have that $\mathsf{det}(\boldsymbol{\Sigma}''_S) \leq \frac{\mathsf{det}(\boldsymbol{\Sigma}')}{\alpha_1 \cdots \alpha_g}$. We therefore need a lower bound on $\alpha_1 \cdots \alpha_g$. Since with all but e^{-11} probability[16], $R_{1,i} \leq (2t +$

[14] For the parameter sets with $n = 131072$, we increase 6.63 below to 7.2 and decrease the probability to e^{-13}.

[15] And for $n = 131072$, $n/2 \cdot e^{-13} \leq 0.15$.

[16] For the parameter sets with $n = 131072$, we increase 6.63 below to 7.2 and decrease the probability to e^{-13}.

$6.63\sqrt{2t}+22) \cdot n\sigma_{h_s}^2$, we have that with probability $1 - n/2 \cdot e^{-11}$ all eigenvalues are greater than

$$L = \frac{\sigma_s^2}{1 + \frac{\sigma_s^2 \cdot (2t + 6.63\sqrt{2t} + 22) \cdot n\sigma_{h_s}^2}{2\sigma_\epsilon^2}}. \tag{14}$$

Combining the above, we have that with at least $p - n \cdot e^{-11}$ probability, all g number of guesses are correct, and

$$\mathsf{det}(\mathbf{\Sigma}'') \leq \frac{\mathsf{det}(\mathbf{\Sigma}')}{L^g}. \tag{15}$$

We note that for the maximum setting of parameters $n = 32768$, $n \cdot e^{-11} \leq 0.55$.[17] Further, $\mathsf{det}(\mathbf{\Sigma}'')$ can be computed by plugging the parameter settings from Sections 8.1 and 8.2 into Lemma 5.1. Thus, we can use (15) to estimate the hardness of the residual instance (after guesses) under a lattice reduction attack.

9 Experiments

9.1 Experimental Set-Up

Parameter sets. We consider the parameter sets proposed by the homomorphicencryption.org standards [2], which were proposed with target security levels of 128, 192 or 256 bits. We update the target estimates using the concrete hardness given by the tool of [21].[18] This is presented in the column "Original Security" in all the tables below. An entry of x/y represents the original target security level x, and y represents the concrete (updated) security level. The standards only consider a ring dimension of up to $n = 32768$, i.e. $\log_2(n) = 15$, but some FHE applications may require a larger ring dimension, up to $\log_2(n) = 17$. We additionally provide estimates for the concrete security of CKKS for values of $\log_2(n) = 17$ by using the parameters given in [33].

Experimental validation. We first provide experimental validation of Lemma 5.1, in Section 9.2. We also provide concrete security estimation for provably secure (statistical) noise-flooding, as presented in [32]. We provide these as a baseline, and to validate our methods. Since there is no reduction in security when applying statistical noise-flooding, those results are presented in [11].

Concrete security experiments set-up. Then, we consider the following experiments. We consider a lattice reduction attack, a guessing attack and a hybrid attack, as outlined in Sections 5, 6 and 7, respectively. We consider these on three types of circuit: the identity circuit, the class of circuits C1 and the class of circuits C2. Recall that these are described in Section 8.

[17] And for $n = 131072$, $n \cdot e^{-13} \leq 0.30$.

[18] Our analysis may give slightly different concrete hardness estimates than the LWE Estimator [4], since [21] takes into account the ellipsoidal distribution of the original secret/error.

Noise-flooding countermeasures. We use the results of [19] to estimate the output variance of the noise ρ^2_{circ}, where circ is one of Identity, C1 or C2. We then consider noise-flooding by ρ^2_{circ}, $100 \cdot \rho^2_{\mathsf{circ}}$ and $t \cdot \rho^2_{\mathsf{circ}}$, for t is the number of decryption queries. For guessing attacks, we do not include results for noise-flooding variance of $t \cdot \rho^2_{circ}$, since in this case, the guessing probability does not go above 10^{-200} for any parameter set. Similarly, for hybrid attacks, we do not include results for noise-flooding variance of $t \cdot \rho^2_{fresh}$, since no coordinates can be guessed with high confidence for any parameter set, and so the attack is equivalent to a lattice reduction attack [19].

9.2 Experimental Validation of Lemma 5.1

We first provide a verification of the theoretical results from Section 5, to demonstrate that the estimations hold in practice. In particular, Lemma 5.1 assumes that the distribution of the coefficients of $\mathbf{e}_1^j$ and $\mathbf{v}^j$ are independent Gaussians, while in practice this is not the case. The quantity of interest is $\det(\mathbf{\Sigma}'^{\sim})$, as defined in Section 5. In the proof of Lemma 5.1, we use the following fact:

$$\det(\mathbf{\Sigma}'^{\sim}) = \frac{\det(\mathbf{H}\mathbf{\Sigma}\mathbf{H}^T + \mathbf{\Sigma}_\varepsilon)}{\det(\mathbf{\Sigma}_\varepsilon)\det(\mathbf{\Sigma})} = \frac{\det\left(\mathbf{I}_{2n} + \frac{1}{\sigma_\epsilon^2}\mathbf{\Sigma}^{1/2}\mathbf{H}\mathbf{H}^T\mathbf{\Sigma}^{1/2}\right)}{\det(\mathbf{\Sigma})}. \tag{16}$$

In order to validate the canonical embedding transformation used in the analysis of Lemma 5.1, we sample a random hint matrix $\mathbf{H}$, directly compute $\mathbf{I}_{2n} + \mathbf{\Sigma}^{1/2}\mathbf{H}\mathbf{H}^T\mathbf{\Sigma}^{1/2}/\sigma_\epsilon^2$, and calculate its determinant. In order to construct the hint matrix, we sample $\mathbf{e}_1^j \leftarrow \chi$ and $\mathbf{v}^j \leftarrow S$ as defined in [11]. We perform this experiment for various settings for the dimension of the LWE secret and error, and for various numbers of hints applied. For each parameter set, we perform 256 trials and take the average of the results in order to compare to the expected value predicted by Lemma 5.1. Figure 2 reports the experimental results, which very closely match the predictions. Notably, we see that the predictions become more accurate as the number of applied hints increases.

We perform this experiment using the SageMath library and run the calculations on an Intel Ice Lake XCC server. Calculating the determinant for larger parameter sets proves computationally infeasible with our experimental setup due to the extreme scaling, as each trial requires multiplying matrices of size $2n \times tn$ and $tn \times 2n$, as well as calculating the determinant of a matrix of size $2n \times 2n$, where n is the dimension and t is the number of hints. Additionally, in order to accurately calculate the final determinant, the numerical values within the matrix require increasingly high floating-point precision (e.g. hundreds or even thousands of bits), further slowing the computation. Our experiments take roughly a week to verify the largest parameter set in Figure 2 ($n = 256$, $t = 16$).

[19] After $\sim$ 200 million decryption queries, the estimated variance does not go below 3.6 for identity circuits, and after $\sim$ 100 million decryption queries does not go below 0.33 and 0.36 for C1 and C2 circuits, respectively.

Dim	Num Hints	Predicted Determinant	Experimental Determinant
64	16	708.60	708.76
64	32	799.19	799.28
64	64	888.87	889.14
64	128	978.08	978.10
64	256	1067.04	1067.00
64	512	1155.89	1155.87
128	16	1594.55	1591.58
128	32	1175.78	1775.55
128	64	1955.17	1954.82
128	128	2133.59	2133.49
128	256	2311.52	2311.44
128	512	2489.22	2489.23
256	16	3543.88	3539.04

Fig. 2: **Summary of results for experimental validation of Lemma 5.1.** Each parameter set is specified by the dimension of the LWE secret/error (column 1) and the number of hints applied (column 2). The third column indicates the (ln of) the expected value of the determinant as predicted by Lemma 5.1. The final column reports the determinant calculated by performing the experiment, as averaged over 256 trials.

9.3 Concrete Security of Lattice Attacks on Identity Circuits

We begin by considering a lattice-reduction attack where the adversary may request any number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See [11]. To calculate the concrete hardness, we apply Lemma 5.1 to obtain the expected volume and dimension of the lattice after hints are integrated and homogenization/isotropization is completed. As in [21], after homogenization/isotropization are performed, the hardness estimates for BKZ require only the volume and dimension of the lattice. These are reported in the final column.

9.4 Concrete Security of Guessing Attacks on Identity Circuits

Next we consider a guessing-only attack, where the adversary may request any number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See [11]. In this attack, the adversary requests enough decryptions so that n LWE secret/error coordinates can be guessed correctly with high probability. Once these coordinates are guessed correctly, the LWE system of equations has a unique solution which can be recovered efficiently using Gaussian elimination. To determine the number of decryptions required to recover the LWE secret/error with some threshold probability, we apply Lemma 6.1 and (7).

9.5 Concrete Security of Hybrid Attacks on Identity Circuits

Here we consider a hybrid attack, where the adversary may request some number of decryptions of fresh ciphertexts (i.e. evaluation of the identity circuit on a fresh ciphertext) with various noise-flooding levels. See [11]. The adversary requests enough decryptions so that some number of LWE secret/error coordinates can be guessed correctly with high probability. The adversary then integrates these guesses into its DBDD instance as perfect hints (as in [21]). Finally, the adversary performs homogenization/isotropization to obtain an SVP instance, and uses a BKZ solver to recover the LWE secret/error. For a fixed number of decryptions, we use (8) to determine the number of guesses g that can be made such that all guesses are correct with high probability. The dimension of the lattice reduces by g, and we compute the volume of the resulting lattice by applying (9). As in [21], after homogenization/isotropization are performed, the hardness estimates for BKZ require only the volume and dimension of the lattice. These are reported in the final column.

9.6 Concrete Security of Lattice Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 9.3, except the adversary requests decryptions of evaluations of a Class 1 or Class 2 circuit (see Sections 8.1 and 8.2) on fresh ciphertexts. To calculate the concrete hardness, we apply Lemma 5.1 to obtain the expected volume and dimension of the lattice after hints are integrated with the parameter settings for $\sigma_{h_s}^2, \sigma_{h_e}^2, \sigma_\epsilon^2$ given in Section 8.1 or Section 8.2 The results are reported in [11].

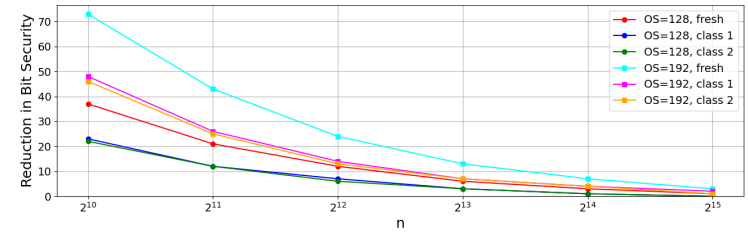
9.7 Concrete Security of Guessing Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 9.4, except the adversary requests decryptions of evaluations of a Class 1 or Class 2 circuit (see Sections 8.1 and 8.2) on fresh ciphertexts. To determine the number of decryptions required to recover the LWE secret with high probability, we apply (12) with the settings of $\sigma_{h_s}, \sigma_{h_e}, \sigma_\epsilon^2$ given in Section 8.1 or Section 8.2. The results for various noise-flooding levels are reported in [11].

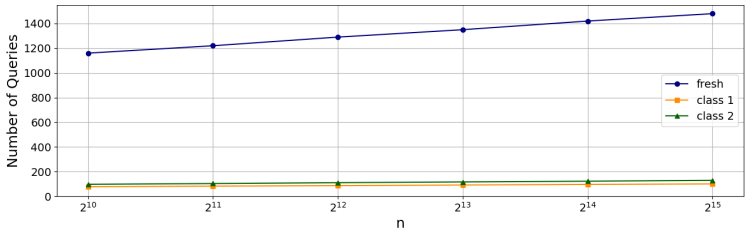
9.8 Concrete Security of Hybrid Attacks on Class 1 and 2 Circuits

This is the same attack as in Section 9.5, except the adversary requests decryptions of evaluations of a Class 1 or Class 2 circuit (see Sections 8.1 and 8.2) on fresh ciphertexts. For a fixed number of decryptions, we use (8), with the settings of $\sigma_{h_2}^2, \sigma_{h_e}^2$, and σ_ϵ^2 given in Section 8.1 or Section 8.2, to determine the number of guesses g that can be made such that all guesses are correct with high probability. The dimension of the lattice reduces by g, and we compute the volume of the resulting lattice by applying (9), with the settings of $\sigma_{h_2}^2, \sigma_{h_e}^2$, and σ_ϵ^2 given in Section 8.1 or Section 8.2. The results are reported in [11].

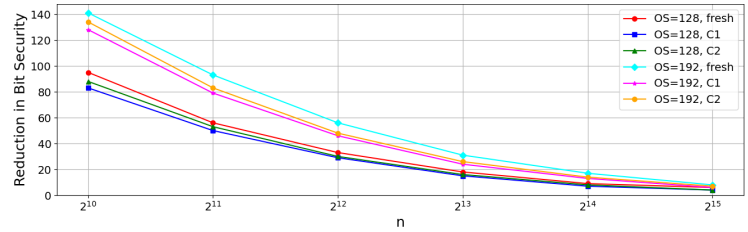
10 Discussion of the Results



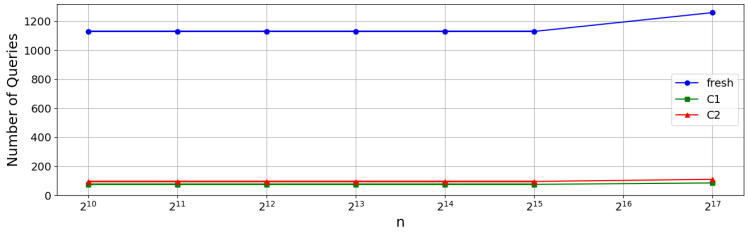
(a) Lattice Reduction Attack



(b) Guessing Attack



(c) Hybrid Attack



(d) Number of Queries in Hybrid Attack

Fig. 3: **Trends for the various attacks.** We compare the efficacy of lattice reduction, guessing, and hybrid attacks for various parameter sets, and for identity, Class 1, and Class 2 circuits with noise-flooding level equal to ρ^2_{fresh}, ρ^2_{C1}, and ρ^2_{C2}, respectively. (a) Shows the reduction in bit security for a lattice reduction attack against an adversary who obtains 1000 decryptions; (b) Shows the number of queries required for guessing n coordinates with probability at least 0.80. (c) Shows the reduction in bit security for a hybrid attack against an adversary who obtains a variable number of decryptions. The number of decryption queries for each parameter set is displayed in (d).

Trends for noise-flooding level of ρ^2_{circ}. Our experimental data is summarized in Figure 3. Figure 3(a) shows the reduction in bit security for a lattice reduction attack when the adversary obtains 1000 decryptions of identity, Class 1, and Class 2 circuits with noise-flooding level ρ^2_{circ} equal to the noise already present in the ciphertext. We note that the graph exhibits a greater reduction in bit-security for identity circuits vs. Class 1 and 2 circuits. We believe the reason is that hints for identity circuits involve all $2n$ coordinates in the LWE secret/error, so the variance of all $2n$ coordinates is reduced after each hint, whereas hints for Class 1 and 2 circuits involve only the n coordinates from the LWE secret, so only the variance of these n coordinates is reduced. We also note that there is a greater security reduction for higher vs. lower target security level. E.g., for the lattice reduction attack, we see that for $\log_2(n) = 10$, identity circuits, and for a security level target of 192, the value of the bit security is reduced by slightly over 70 bits. On the other hand, for the same circuit, target security level, and attack, for $\log_2(n) = 15$, bit security reduction is less than 5 bits. In fact, the reduction in security seems highly correlated with decrease in modulus. When fixing the dimension n, target security level of 192 have smaller modulus q_L, compared to target security level of 128 and as the modulus q_L becomes smaller, "hints" obtained from decryption have more of an impact on the bit-security for lattice reduction attacks. The same trends can be seen in the Hybrid attack.

Figure 3(b) shows the number of queries required for guessing n coordinates with high probability for identity, Class 1 and Class 2 circuits. We note that guessing attacks perform significantly better for Class 1 and 2 circuits versus identity circuits. For identity circuits, there are a total of $2n$ eigenvalues that are reduced by obtaining hints, but n of these eigenvalues have relatively larger expectation, while n have smaller expectation (this is because for identity circuits, hints correspond to linear combinations of both the s and e variables, in which the s variables have variance $2/3$, while the e variables have variance 3.2^2). The eigenvectors corresponding to these eigenvalues do not align with the standard basis. Therefore, for purposes of fast estimates, we only take into account the trace (i.e. sum of the eigenvalues) and, given trace T, we argue that the variance of the n secret or error coordinates with smallest variance is at most $\mathsf{T}/(2n)$. However, in practice, the n coordinates with the smallest variance may have variance significantly smaller than $\mathsf{T}/(2n)$. For Class 1 and 2 circuits, hints correspond to linear combinations of *only* the s variables from the LWE instance. Thus, we restrict our attention to a subspace with only n eigenvalues. All of these eigenvalues have the same distribution, and our proof shows that *all* the eigenvalues are less than maximum value σ^2_{max}.

Figure 3(c) shows the reduction in bit-security for a hybrid attack when considering an adversary who obtains decryptions of identity, Class 1, and Class 2 circuits. Figure 3(d) shows the number of queries obtained in each of these attacks. We chose the number of queries for the identity, Class 1, and Class 2 circuits so that a significant number of guesses can be made for each parameter set (otherwise the attack will be very similar to a lattice reduction attack). Based on the discussion above, this means that the number of queries required is far

higher for identity circuits than Class 1 and Class 2 circuits. Thus, after guesses are made, the residual instance has lower variance in the case of identity circuits (since more hints have been incorporated, with each hint slightly reducing the variance). This explains why for approximately the same number of guesses, the reduction in bit-security is greater for identity circuits versus Class 1 and Class 2 circuits, as can be observed from the graph.

Trends across various noise-flooding levels. In the full version [11] we validate that there is no security drop in our experiments when using the statistically-secure noise-flooding levels proposed in [32]. Indeed, we observed *no* reduction in either the security level or in the bikz for any parameter setting. Recall that we investigate the effectiveness of noise-flooding levels ρ^2_{circ}, $100 \cdot \rho^2_{\mathsf{circ}}$, and $t \cdot \rho^2_{\mathsf{circ}}$, where t is the number of decryption queries, circ is one of Identity, C1 or C2, and ρ^2_{circ} is the noise variance present in the ciphertext. As expected, we see that the biggest drop in bit security is observed when noise-flooding by ρ^2_{circ}, across all parameter sets and across all circuits. In contrast, we observe that noise-flooding by $t \cdot \rho^2_{\mathsf{circ}}$ leads to a very low reduction in the security level, if at all. As opposed to a 70-bit security drop seen for lattice attacks with $\log_2(n) = 10$ and 192-bit security for identity circuits with noise level ρ^2_{fresh}, when noise-flooding by $t \cdot \rho^2_{\mathsf{fresh}}$, the security level drops by only a few bits. Further, as the value of $\log_2(n)$ (and thus also q_L increases), the security level drop decreases. For example, for $\log_2(n) = 17$, there is no change in the security level.

Conclusions: We observe that, in practice, there is essentially no reduction in security when noise flooding with variance $t \cdot \rho^2_{\mathsf{circ}}$, where t is the number of decryption queries, and ρ^2_{circ} is the noise variance, as predicted by an average-case noise analysis. One may also consider noise flooding by $\alpha \cdot t \cdot \rho$, where $0 < \alpha < 1$, if it is acceptable for the security level to drop by a few bits. There is no definitive setting of α which is "best," and one can think of α as a parameter to be fine-tuned depending on the application. In particular, α can be adjusted to allow for more decryption queries, or to reduce the message precision loss, since noise-flooding by $x \cdot \rho^2_{\mathsf{circ}}$ incurs an *additional* loss of $\frac{1}{2}\log_2(x+1)$ bits in the message precision. Finally, we note that the techniques developed in this paper, as well as the experimental results presented, can be used as a way to establish key refreshing policies in a concrete application. E.g., if the noise level is set to $\alpha \cdot t \cdot \rho$, the keys should be refreshed after releasing t number of decryptions. Thus, there is a tradeoff among frequency of key refresh, an acceptable precision loss, and an acceptable drop in bit-security.

Acknowledgements

We thank the anonymous reviewers, Tikaram Sanyashi, and Alexander Viand for helpful comments and discussions. Anamaria Costache is supported in part by Intel through the Intel Labs Soteria Research Collaboration. Dana Dachman-Soled, Rui Tang, and Hunter Kippen are supported in part by NSF grant #CNS-2154705 and by Intel through the Intel Labs Crypto Frontiers Research Center. Hunter Kippen is supported in part by the Clark Doctoral Fellowship from the Clark School of Engineering, University of Maryland, College Park.

References

1. Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, et al. Openfhe: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 53–63, 2022.
2. Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. homomorphicencryption.org, 2018.
3. Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 297–322, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland.
4. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015.
5. Andreea Alexandru, Ahmad Al Badawi, Daniele Micciancio, and Yuriy Polyakov. Application-aware approximate homomorphic encryption: Configuring fhe for practical use. Cryptology ePrint Archive, Paper 2024/203, 2024. https://eprint.iacr.org/2024/203.
6. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association.
7. Private Communication, anonymized for submission.
8. Diego F. Aranha, Anamaria Costache, Antonio Guimarães, and Eduardo Soria-Vazquez. Heliopolis: Verifiable computation over homomorphically encrypted data from interactive oracle proofs is practical. Cryptology ePrint Archive, Paper 2023/1949, 2023. https://eprint.iacr.org/2023/1949.
9. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 483–501, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany.
10. Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 201–218, Zurich, Switzerland, February 9–11, 2010. Springer Berlin Heidelberg, Germany.
11. Flavio Bergamaschi, Anamaria Costache, Dana Dachman-Soled, Hunter Kippen, Lucas LaBuff, and Rui Tang. Revisiting the security of approximate FHE with noise-flooding countermeasures. Cryptology ePrint Archive, Paper 2024/424, 2024.
12. Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in*

Computer Science, pages 565–596, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Cham, Switzerland.
13. Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. Cryptology ePrint Archive, Report 2018/822, 2018.
14. Katharina Boudgoust and Peter Scholl. Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part I*, volume 14438 of *Lecture Notes in Computer Science*, pages 371–404, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.
15. Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto. Attacks against the IND-CPA$^{\mathrm{D}}$ security of exact FHE schemes. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 2505–2519, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press.
16. Jung Hee Cheon, Seungwan Hong, and Duhyeong Kim. Remark on the security of CKKS scheme in practice. Cryptology ePrint Archive, Report 2020/1581, 2020.
17. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland.
18. Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. Efficient threshold FHE with application to real-time systems. Cryptology ePrint Archive, Report 2022/1625, 2022.
19. Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *SAC 2023: 30th Annual International Workshop on Selected Areas in Cryptography*, volume 14201 of *Lecture Notes in Computer Science*, pages 325–345, Fredericton, Canada, August 14-18, 2024. Springer, Cham, Switzerland.
20. Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and tradeoffs for HElib. In Mike Rosulek, editor, *Topics in Cryptology – CT-RSA 2023*, volume 13871 of *Lecture Notes in Computer Science*, pages 29–53, San Francisco, CA, USA, April 24–27, 2023. Springer, Cham, Switzerland.
21. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland.
22. Dana Dachman-Soled, Huijing Gong, Tom Hanson, and Hunter Kippen. Revisiting security estimation for LWE with hints from a geometric perspective. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 748–781, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.
23. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315, Santa Barbara, CA, USA, August 20–24, 1990. Springer, New York, USA.

24. Sanjam Garg, Aarushi Goel, and Mingyuan Wang. How to prove statements obliviously? Cryptology ePrint Archive, Paper 2023/1609, 2023. https://eprint.iacr.org/2023/1609.
25. Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482, Santa Barbara, CA, USA, August 15–19, 2010. Springer Berlin Heidelberg, Germany.
26. Timo Glaser, Alexander May, and Julian Nowakowski. Entropy suffices for guessing most keys. Cryptology ePrint Archive, Paper 2023/797, 2023. https://eprint.iacr.org/2023/797.
27. Qian Guo, Denis Nabokov, Elias Suvanto, and Thomas Johansson. Key recovery attacks on approximate homomorphic encryption with non-worst-case noise flooding countermeasures. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024: 33rd USENIX Security Symposium*, Philadelphia, PA, USA, August 14–16, 2024. USENIX Association.
28. Suk-Geun Hwang. Cauchy's interlace theorem for eigenvalues of hermitian matrices. *American Mathematical Monthly*, pages 157–159, 2004.
29. Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 549–580, Santa Barbara, CA, USA, August 20–24, 2023. Springer, Cham, Switzerland.
30. Rafał Latała and Dariusz Matlak. Royen's proof of the gaussian correlation inequality. In *Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2014–2016*, pages 265–275. Springer, 2017.
31. Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 648–677, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland.
32. Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 560–589, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland.
33. Oliver Masters, Hamish Hunt, Enrico Steffinlongo, Jack Crawford, Flavio Bergamaschi, Maria E. Dela Rosa, Caio C. Quini, Camila T. Alves, Feranda de Souza, and Deise G. Ferreira. Towards a homomorphic machine learning big data pipeline for the financial services sector. Cryptology ePrint Archive, Report 2019/1113, 2019.
34. Alexander May and Julian Nowakowski. Too many hints - when LLL breaks LWE. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part IV*, volume 14441 of *Lecture Notes in Computer Science*, pages 106–137, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.
35. Daniele Micciancio and Adam Suhl. Simulation-secure threshold PKE from LWE with polynomial modulus. Cryptology ePrint Archive, Report 2023/1728, 2023.